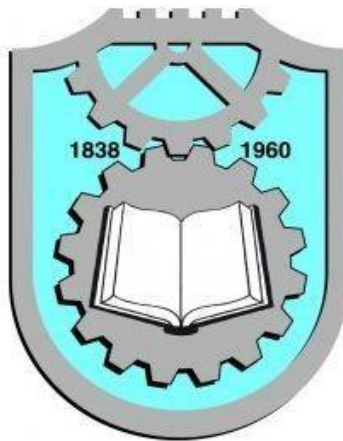


**UNIVERZITET U KRAGUJEVCU**  
**FAKULTET INŽENJERSKIH NAUKA**



**Veštačka inteligencija**

**Dokumentacija za projektni zadatak**

**Klasifikacija tekstura**

Student:  
Nikola Džajević 615/2017

Profesor:  
Dr Vesna Ranković  
Saradnik:  
Tijana Šušteršić

*Kragujevac, 2020/21.*

## **Sadržaj**

1. Uvod .....	3
2. Obrada i priprema podataka .....	4
3. Neuronska mreža .....	27
3.1 Priprema radnog okruženja .....	28
3.2 Kreiranje neuronske mreže .....	28
3.3 Treniranje i testiranje neuronske mreže .....	29
3.4 Rezultati neuronske mreže .....	29
4. Zaključak.....	34
Literatura .....	35

## 1. Uvod

Tema projektnog zadatka je treniranje neuronske mreže tako da razlikuje između 11 tekstura. <sup>(2)</sup>

Neuronska mreža je jedan oblik implementacije sistema veštačke inteligencije, koji predstavlja sistem koji se sastoji od određenog broja međusobno povezanih procesora ili čvorova koje nazivamo neuronima. Svaki neuron ima lokalnu memoriju u kojoj pamti podatke koje obrađuje, i ti podaci su uglavnom numerički. Da bi se stvorila neuronska mreža, potrebno je kreirati njenu arhitekturu, odnosno specifično povezivanje neurona u jednu celinu. Struktura neuronske mreže se razlikuje po broju slojeva i najčešće ih ima tri. Prvi sloj se naziva ulazni, poslednji sloj predstavlja izlazni, dok se slojevi između nazivaju skriveni slojevi (*hidden layers*). Ulazni sloj prima podatke iz spoljašnje sredine, skriveni slojevi prosleđuju relevantne podatke izlaznom sloju koji na kraju daje konačni rezultat. <sup>(1)</sup>



Slika 1 – Grafički prikaz tekstura

Bitno je samo napomenuti da složenije neuronske mreže imaju više skrivenih slojeva i da su svi slojevi međusobno potpuno povezani. Komunikacija između slojeva je takva da se izlaz svakog neurona iz prethodnog sloja povezuje sa ulazima svih neurona narednog sloja i jačina tih veza se naziva težinski faktor (*weight*).

Kratka podela odnosno klasifikacija neuronskih mreža može biti slededa:

1. Prema broju slojeva
  - a. Jednoslojne
  - b. Višeslojne
2. Vrsti veza između neurona
3. Vrsti obučavanja neuronskih mreža
4. Smeru prostiranja informacija
  - a. Nepovratne (*Feedforward*)
  - b. Povratne (*Feedback*)
5. Vrsti podataka

Neuronska mreža se može realizovati na dva načina:

1. Hardverska realizacija (fizičko povezivanje čvorova)
2. Softverska realizacija (logičko – virtuelno povezivanje slojeva)

U projektu je korišćena višeslojna neuronska mreža nepovratnog tipa. Ostali parametri mreže biće detaljno opisani u nastavku rada.

## 2. Obrada i priprema podataka

Odmah na početku je potrebno obraditi ulazne podatke i ujedno ih spremiti za dalju primenu – rad sa njima. Kako je i gore navedeno, potrebno je na osnovu određenih podataka (parametara) istrenirati neuronsku mrežu tako da razlikuje između 11 različitih tekstura.

Texture data set			
Type	Classification	Origin	Real world
Features	40	(Real / Integer / Nominal)	(40 / 0 / 0)
Instances	5500	Classes	11
Missing values?			No

Slika 2 – Pregled seta podataka

Podaci su sastavljeni od 40 atributa (parametra) kao i 11 mogućih klasa tekstura. Podataka ukupno ima 5500 , pri tom nemamo podatke bez vrednosti. Važno je samo napomenuti da su podaci ‘preuzeti’ iz realnog sveta.<sup>(3)</sup>

Atributi koji su dati unutar ovog seta podataka su sledeći:

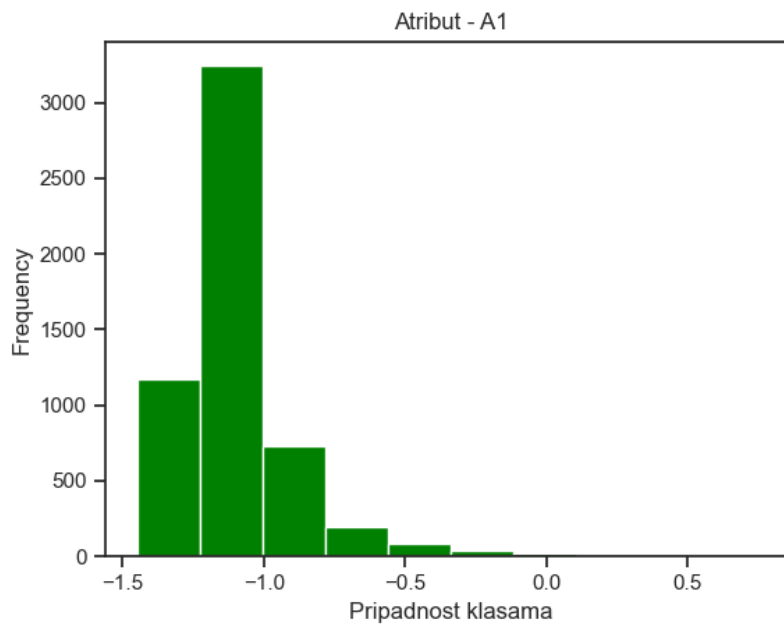
1. Atribut - A1
  2. Atribut – A2
  3. Atribut – A3
  4. Atribut – A4
  5. Atribut – A5
  6. Atribut – A6
  7. Atribut – A7
  8. Atribut – A8
  9. Atribut – A9
  10. Atribut – A10
  11. Atribut – A11
  12. Atribut – A12
  13. Atribut – A13
  14. Atribut – A14
  15. Atribut – A15
  16. Atribut – A16
  17. Atribut – A17
  18. Atribut – A18
  19. Atribut – A19
  20. Atribut – A20
  21. Atribut – A21
  22. Atribut – A22
  23. Atribut – A23
  24. Atribut – A24
  25. Atribut – A25
  26. Atribut – A26
  27. Atribut – A27
  28. Atribut – A28
  29. Atribut – A29
  30. Atribut – A30
  31. Atribut – A31
  32. Atribut – A32
  33. Atribut – A33
  34. Atribut – A34
  35. Atribut – A35
  36. Atribut – A36
  37. Atribut – A37
  38. Atribut – A38
  39. Atribut – A39
  40. Atribut – A40
-

Svaki od atributa ima svoja predefinisana moguća stanja – vrednosti. U nastavku se mogu videti sva moguća stanja (vrednosti) za svaki atribut pojedinačno.

Attribute	Domain	Attribute	Domain	Attribute	Domain
A1	[-1.45, 0.774]	A15	[-0.943, 0.164]	A28	[-1.154, 0.422]
A2	[-1.2, 0.33]	A16	[-0.994, 0.036]	A29	[-1.132, 0.392]
A3	[-1.31, 0.344]	A17	[-1.172, 0.02]	A30	[-1.422, 0.472]
A4	[-1.11, 0.588]	A18	[-1.017, 0.115]	A31	[-1.45, 0.774]
A5	[-1.053, 0.439]	A19	[-1.004, 0.083]	A32	[-1.179, 0.565]
A6	[-1.003, 0.452]	A20	[-1.18, 0.439]	A33	[-1.147, 0.675]
A7	[-1.208, 0.525]	A21	[-1.45, 0.774]	A34	[-1.123, 0.313]
A8	[-1.08, 0.398]	A22	[-1.228, 0.596]	A35	[-1.015, 0.34]
A9	[-1.057, 0.437]	A23	[-1.341, 0.446]	A36	[-1.03, 0.156]
A10	[-1.258, 0.355]	A24	[-1.177, 0.688]	A37	[-1.253, 0.09]
A11	[-1.45, 0.774]	A25	[-1.137, 0.41]	A38	[-1.097, 0.194]
A12	[-1.083, 0.372]	A26	[-1.11, 0.373]	A39	[-1.076, 0.202]
A13	[-1.119, 0.635]	A27	[-1.239, 0.612]	A40	[-1.215, 0.465]
A14	[-1.018, 0.157]	Class	{2, 3, 4, 9, 10, 7, 6, 8, 12, 13, 14}		

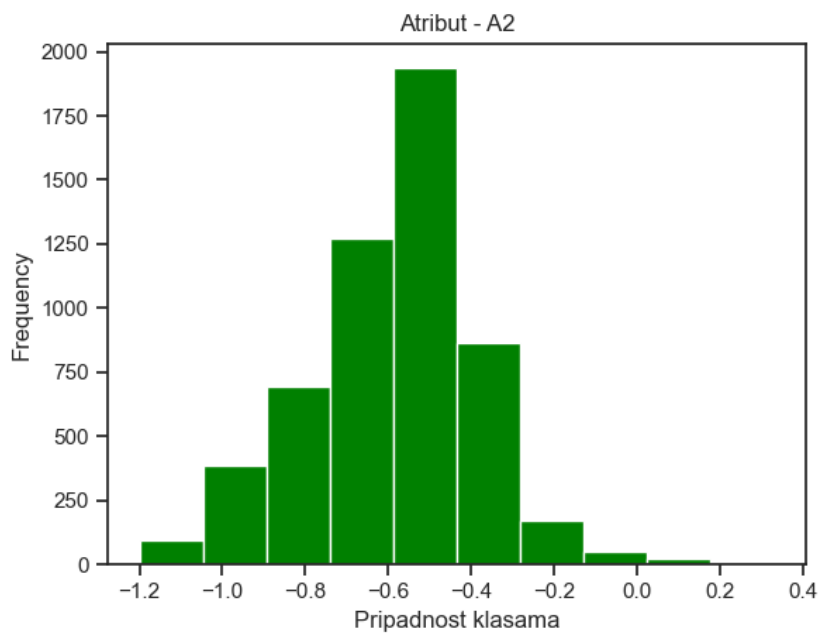
Slika 3 – Pregled atributa i njegovih mogućih vrednosti

- Atribut br. 1 – A1
  - Opseg vrednosti [-1.45, 0.774]



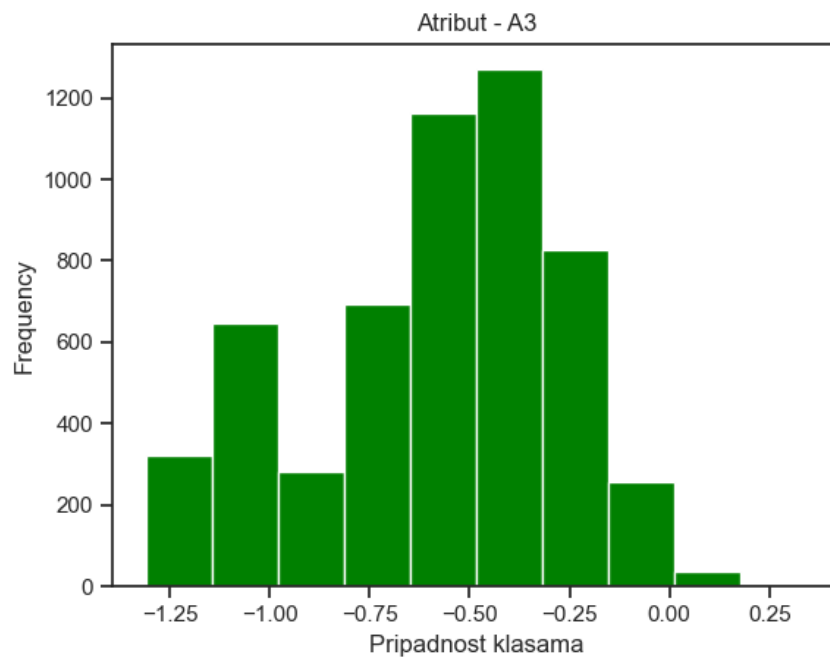
Slika 4 – Grafički prikaz atributa A1

- Atribut br. 2 – A2
  - Opseg vrednosti [-1.2, 0.33]



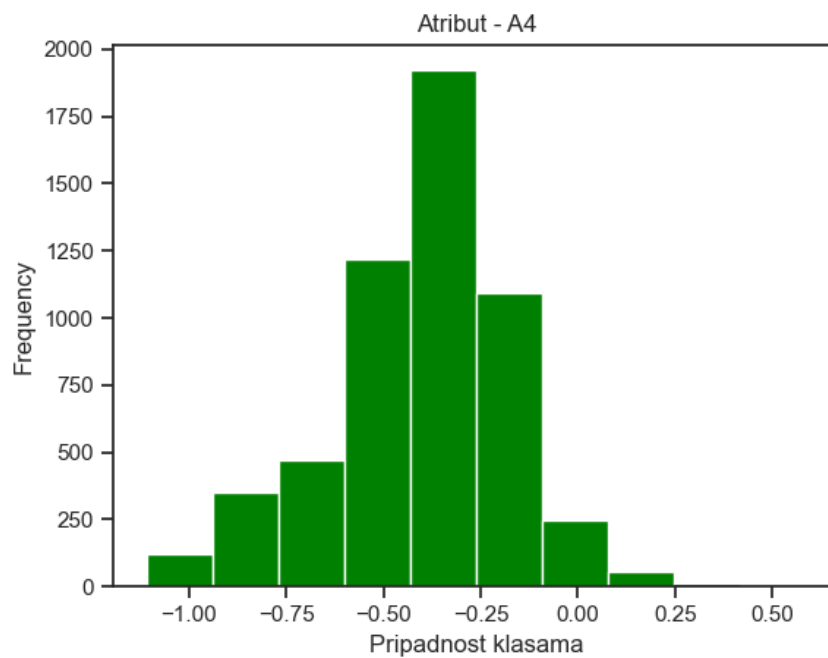
Slika 5 – Grafički prikaz atributa A2

- Atribut br. 3 – A3
  - Opseg vrednosti [-1.31, 0.344]



Slika 6 – Grafički prikaz atributa A3

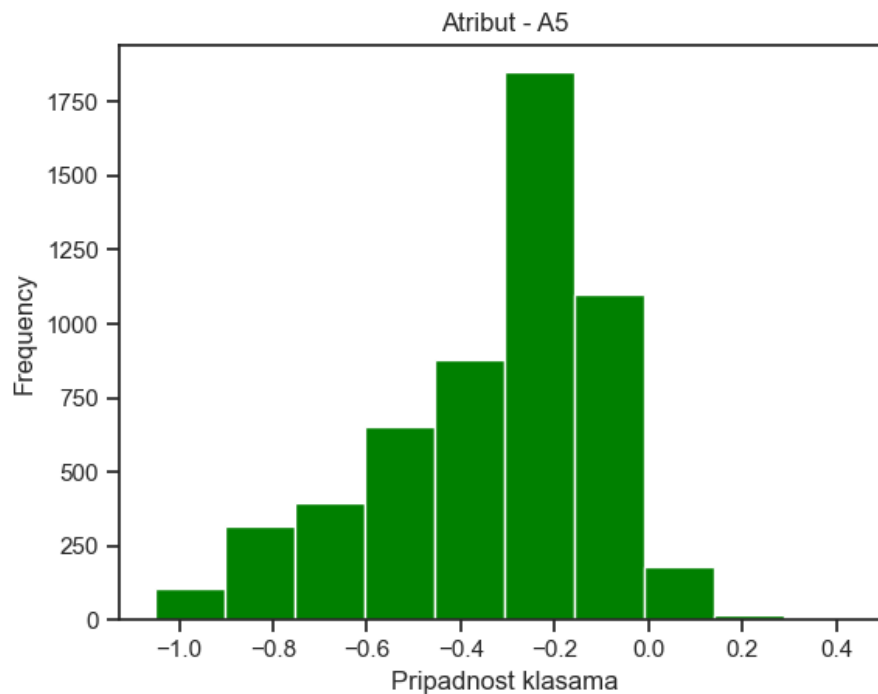
- Atribut br. 4 – A4
  - Opseg vrednosti [-1.11, 0.588]



Slika 7 – Grafički prikaz atributa A4

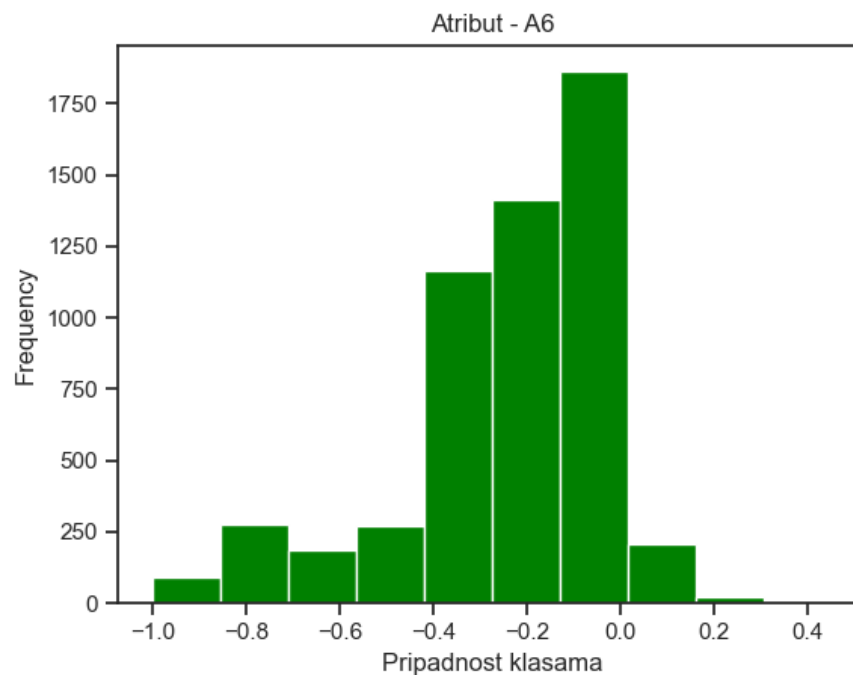


- Atribut br. 5 – A5
  - Opseg vrednosti [-1.053, 0.439]



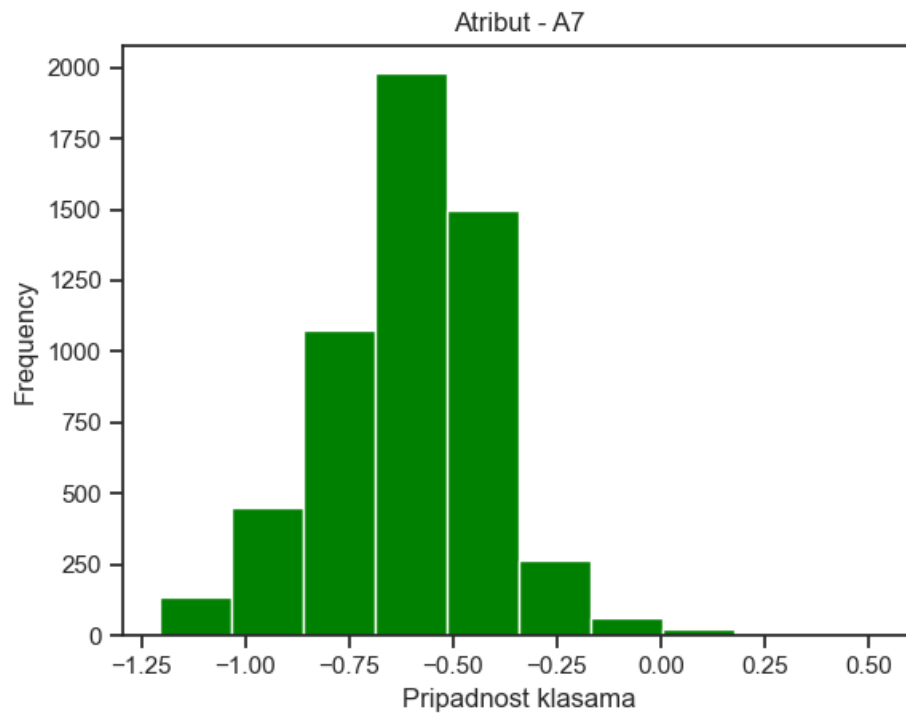
Slika 8 – Grafički prikaz atributa A5

- Atribut br. 6 – A6
  - Opseg vrednosti [-1.003, 0.452]



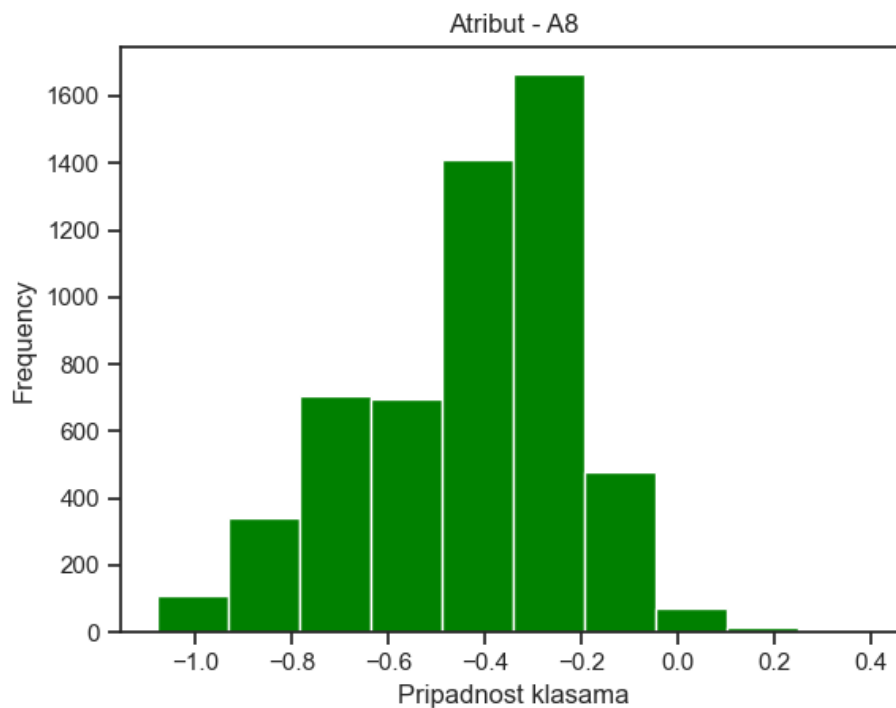
Slika 9 – Grafički prikaz atributa A6

- Atribut br. 7 – A7
  - Opseg vrednosti [-1.208, 0.525]



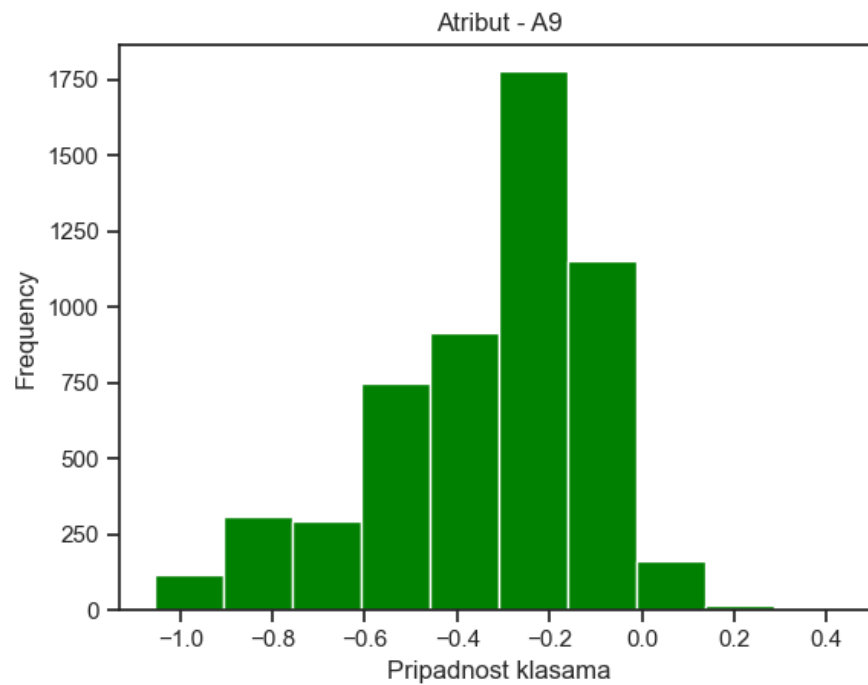
Slika 10 – Grafički prikaz atributa A7

- Atribut br. 8 – A8
  - Opseg vrednosti [-1.08, 0.398]



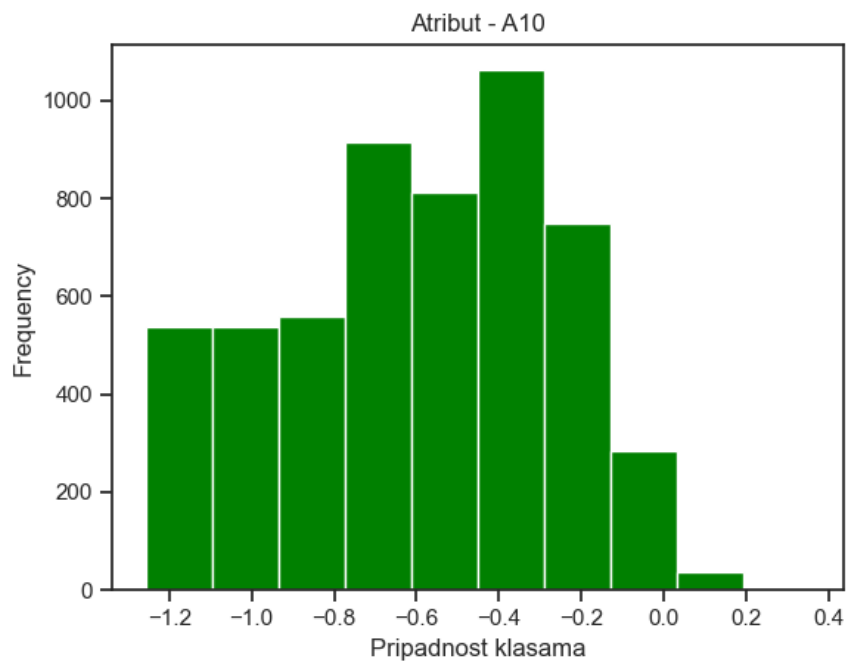
Slika 11 – Grafički prikaz atributa A8

- Atribut br. 9 – A9
  - Opseg vrednosti [-1.057, 0.437]



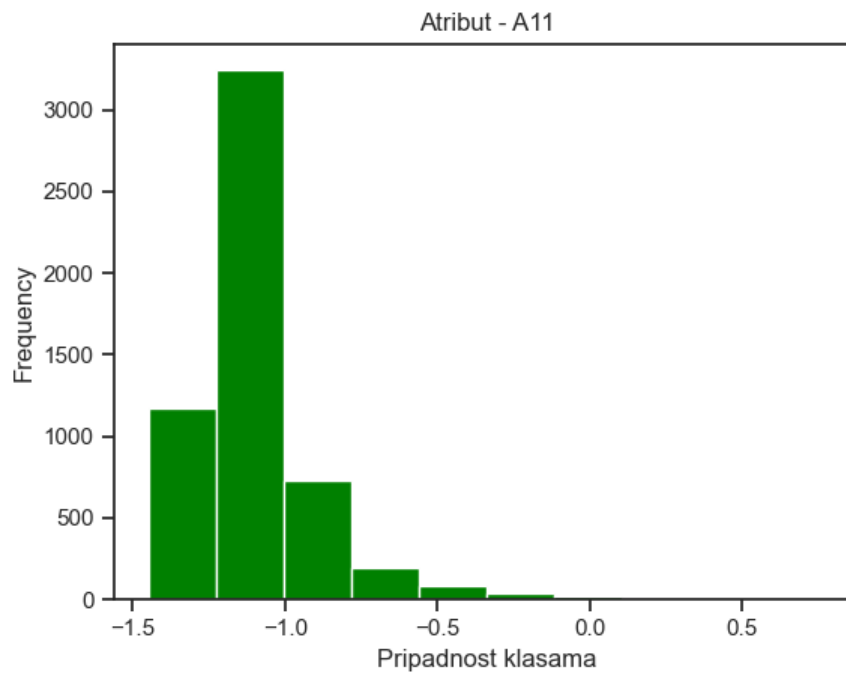
Slika 12 – Grafički prikaz atributa A9

- Atribut br. 10 – A10
  - Opseg vrednosti [-1.258, 0.355]



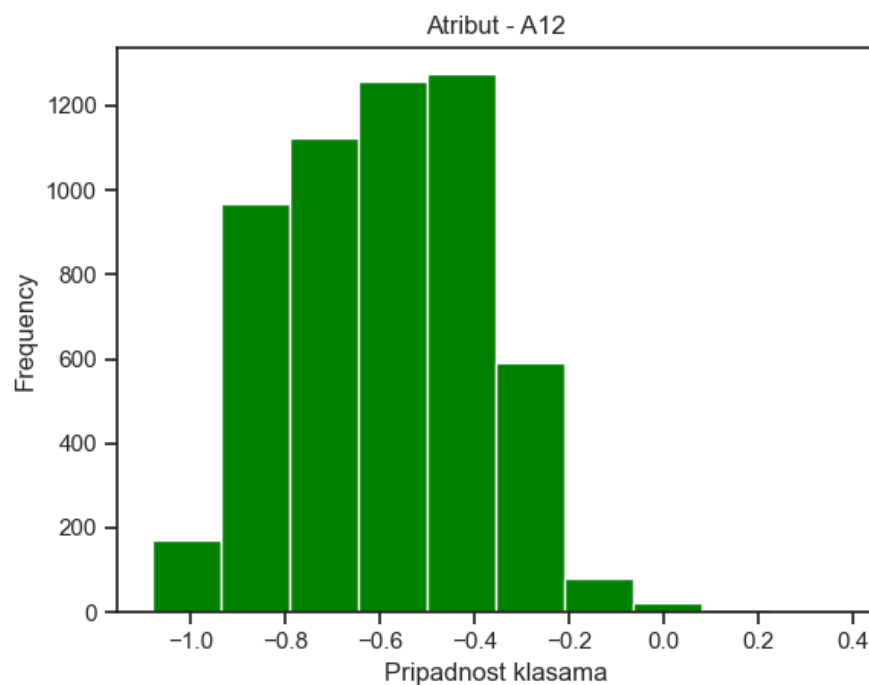
Slika 13 – Grafički prikaz atributa A10

- Atribut br. 11 – A11
  - Opseg vrednosti [-1.45, 0.774]



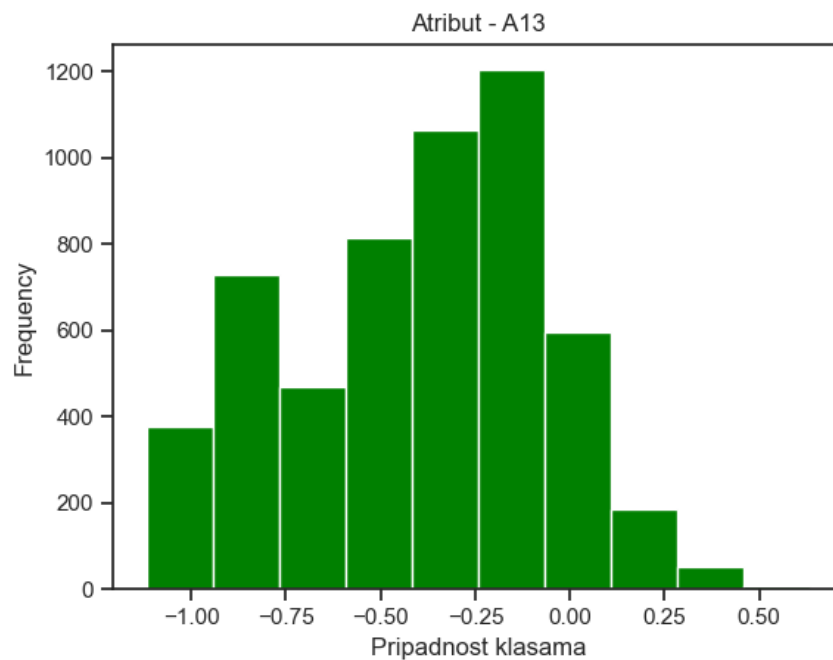
Slika 14 – Grafički prikaz atributa A11

- Atribut br. 12 – A12
  - Opseg vrednosti [-1.083, 0.372]



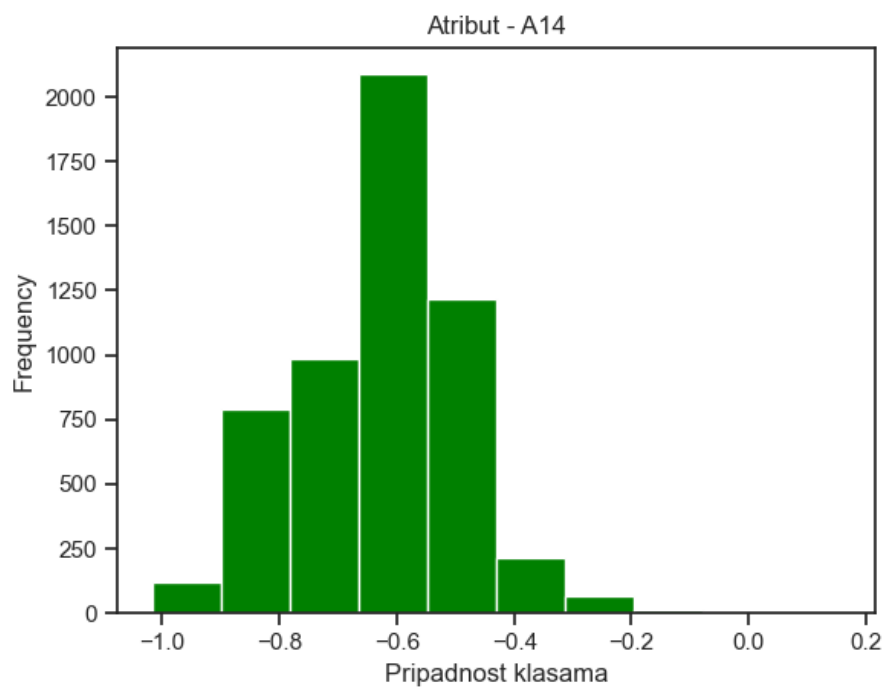
Slika 15 – Grafički prikaz atributa A12

- Atribut br. 13 – A13
  - Opseg vrednosti [-1.119, 0.635]



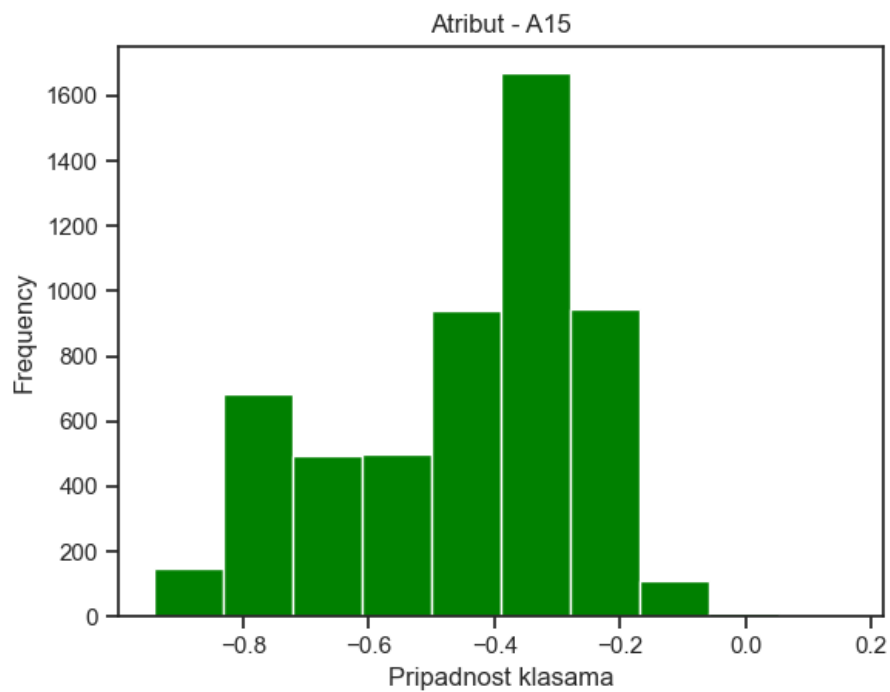
Slika 16 – Grafički prikaz atributa A13

- Atribut br. 14 – A14
  - Opseg vrednosti [-1.018, 0.157]



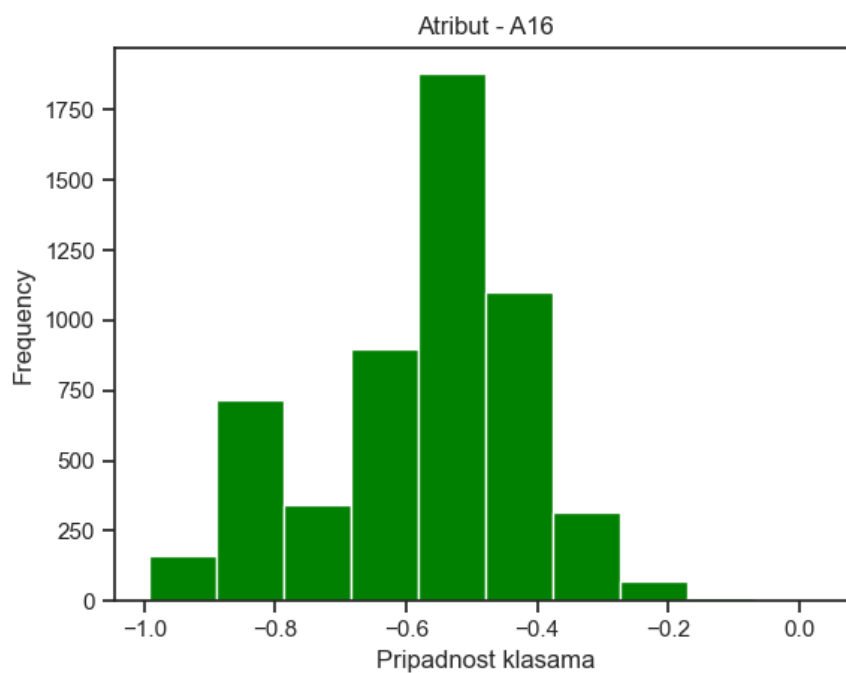
Slika 17 – Grafički prikaz atributa A14

- Atribut br. 15 – A15
  - Opseg vrednosti [-0.943, 0.164]



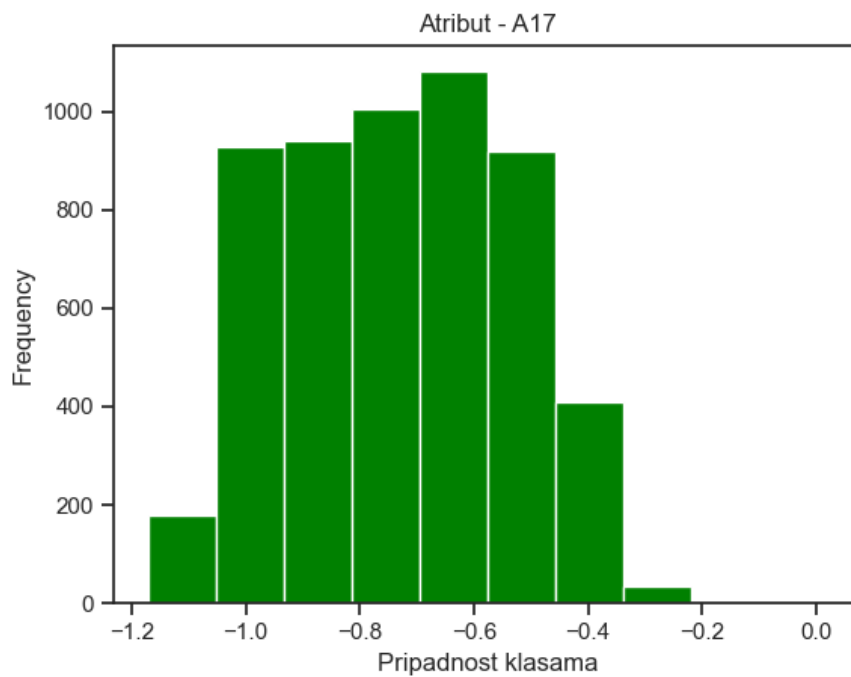
Slika 18 – Grafički prikaz atributa A15

- Atribut br. 16 – A16
  - Opseg vrednosti [-0.994, 0.036]



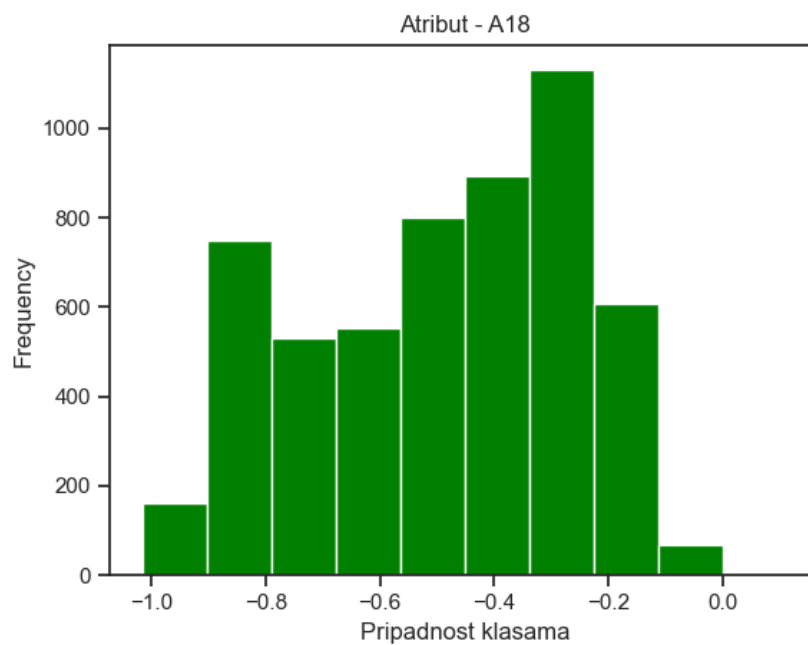
Slika 19 – Grafički prikaz atributa A16

- Atribut br. 17 – A17
  - Opseg vrednosti [-1.172, 0.02]



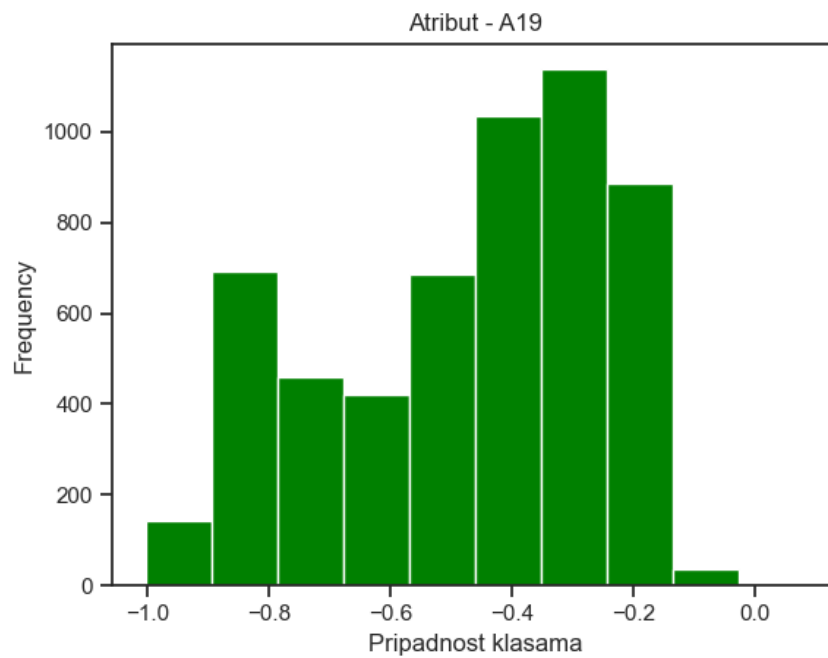
Slika 20 – Grafički prikaz atributa A17

- Atribut br. 18 – A18
  - Opseg vrednosti [-1.017, 0.115]



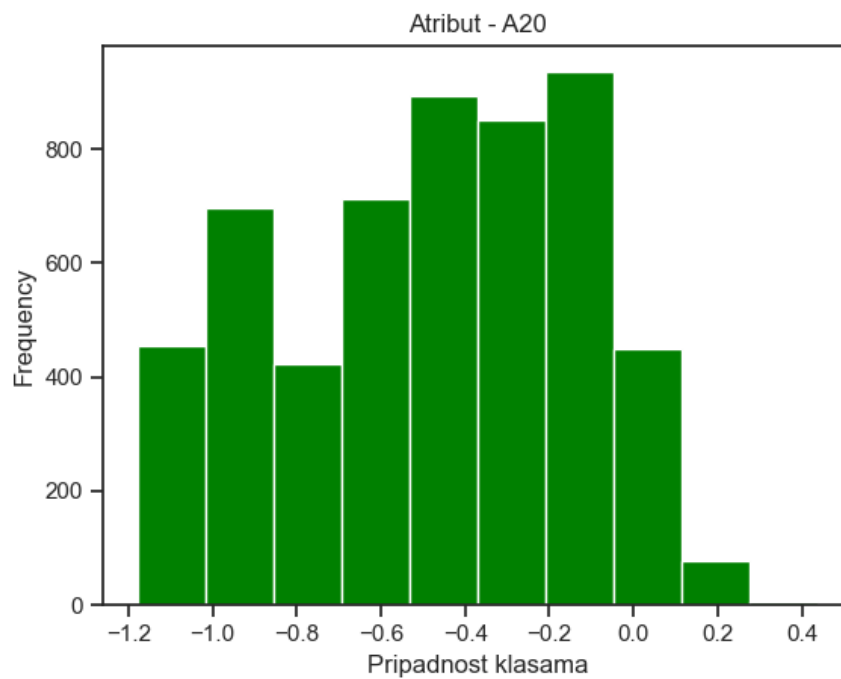
Slika 21 – Grafički prikaz atributa A18

- Atribut br. 19 – A19
  - Opseg vrednosti [-1.004, 0.083]



Slika 22 – Grafički prikaz atributa A19

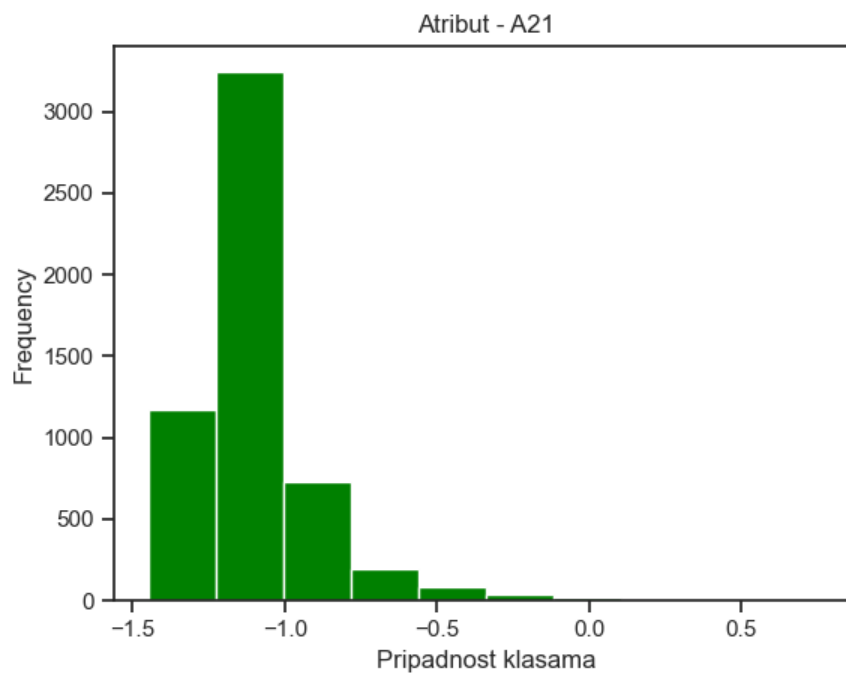
- Atribut br. 20 – A20
  - Opseg vrednosti [-1.18, 0.439]



Slika 23 – Grafički prikaz atributa A20

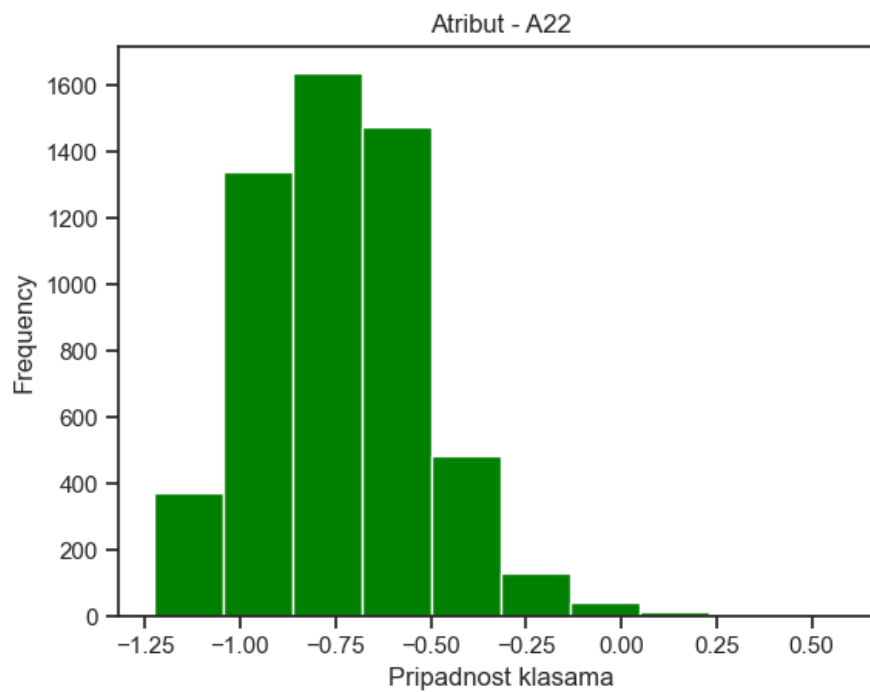


- Atribut br. 21 – A21
  - Opseg vrednosti [-1.45, 0.774]



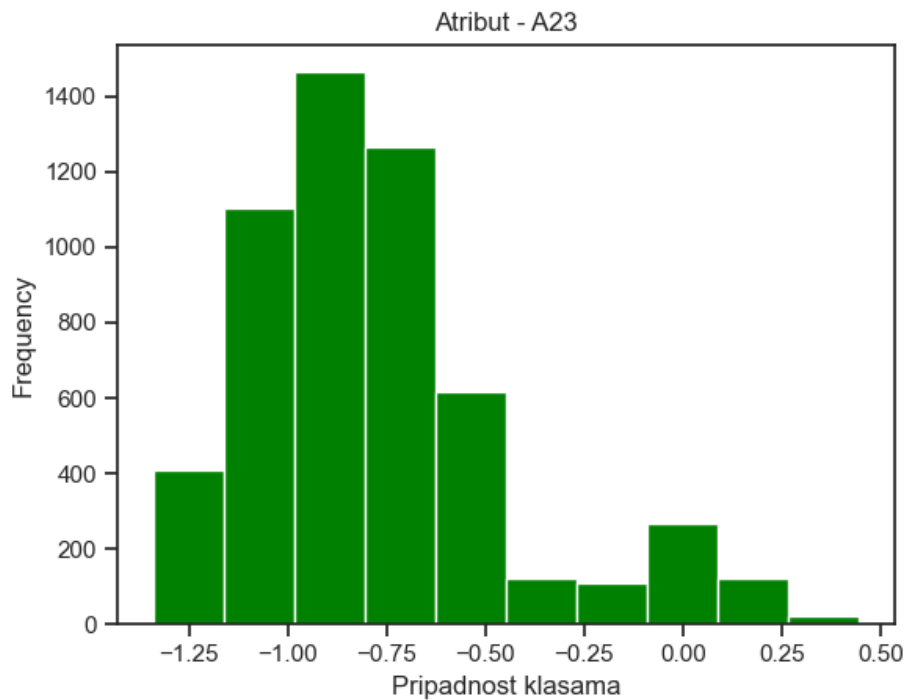
Slika 24 – Grafički prikaz atributa A21

- Atribut br. 22 – A22
  - Opseg vrednosti [-1.228, 0.596]



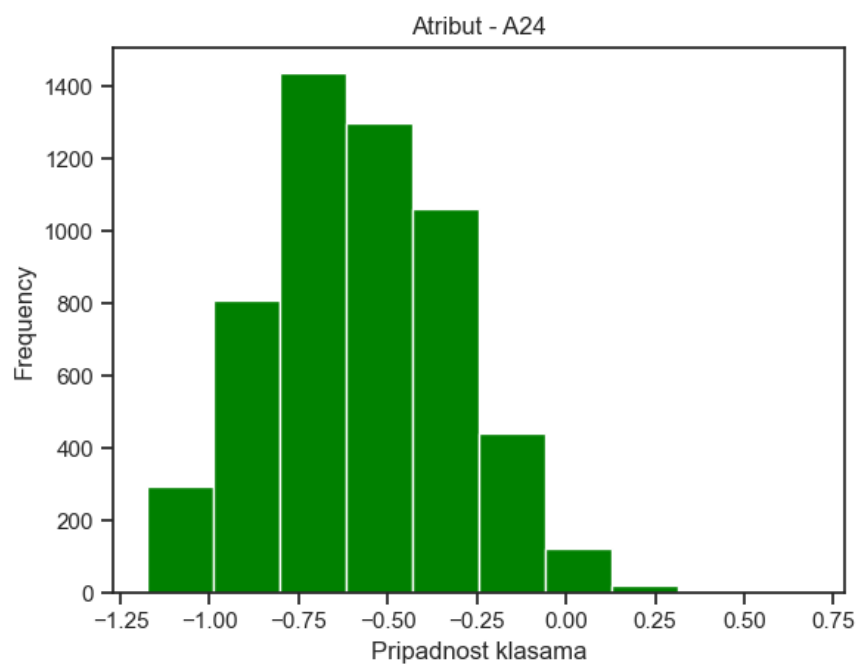
Slika 25 – Grafički prikaz atributa A22

- Atribut br. 23 – A23
  - Opseg vrednosti [-1.341, 0.446]



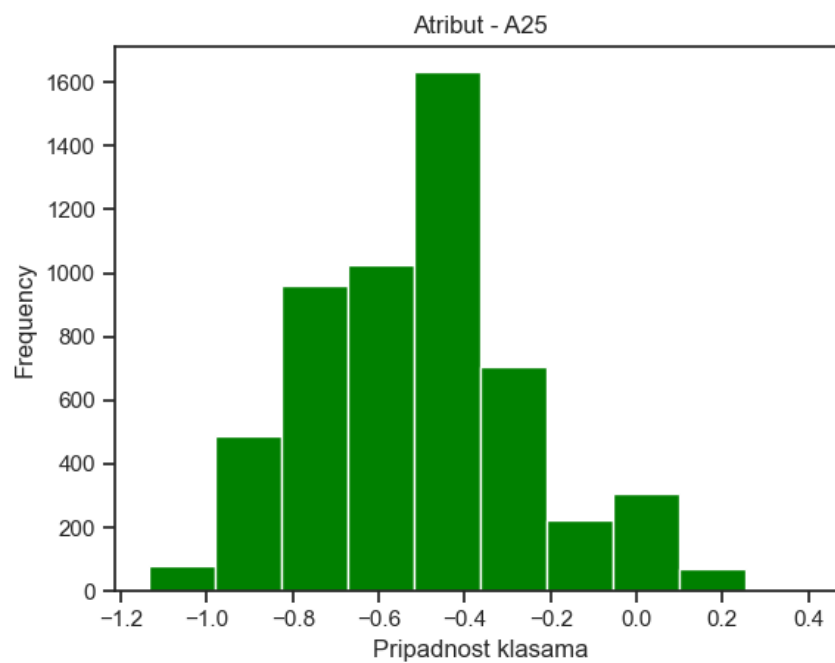
Slika 26 – Grafički prikaz atributa A23

- Atribut br. 24 – A24
  - Opseg vrednosti [-1.177, 0.688]



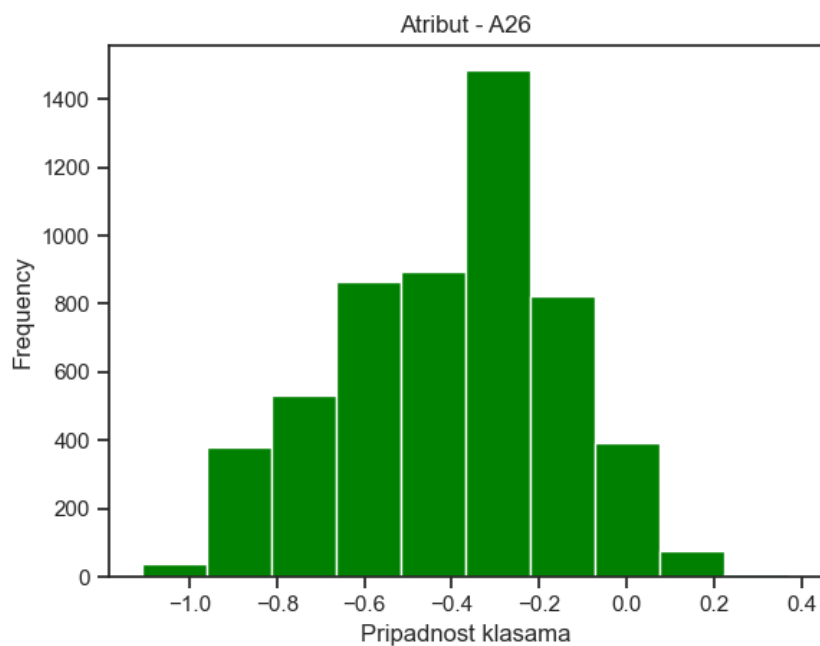
Slika 27 – Grafički prikaz atributa A24

- Atribut br. 25 – A25
  - Opseg vrednosti [-1.137, 0.41]



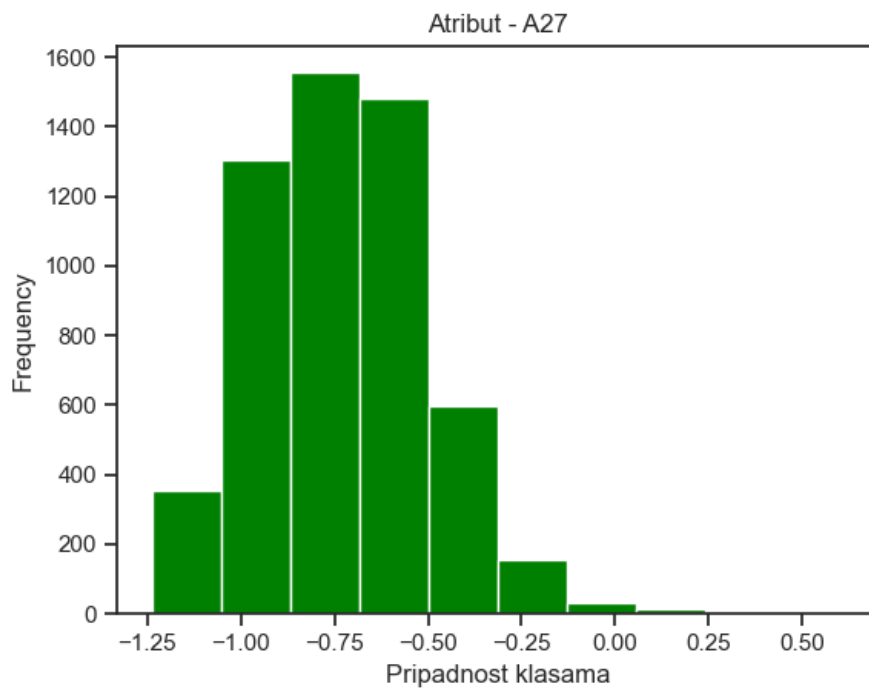
Slika 28 – Grafički prikaz atributa A25

- Atribut br. 26 – A26
  - Opseg vrednosti [-1.11, 0.373]



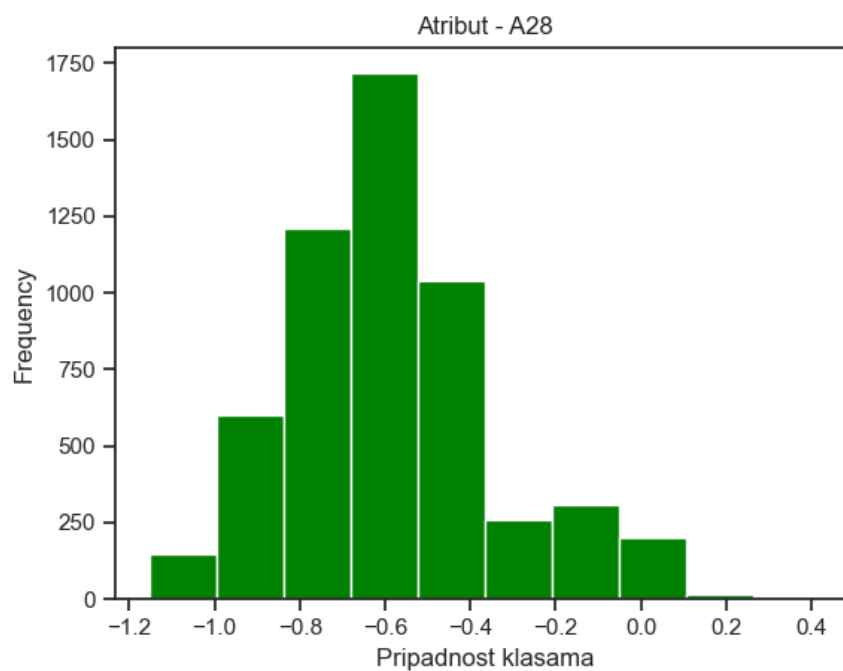
Slika 29 – Grafički prikaz atributa A26

- Atribut br. 27 – A27
  - Opseg vrednosti [-1.239, 0.612]



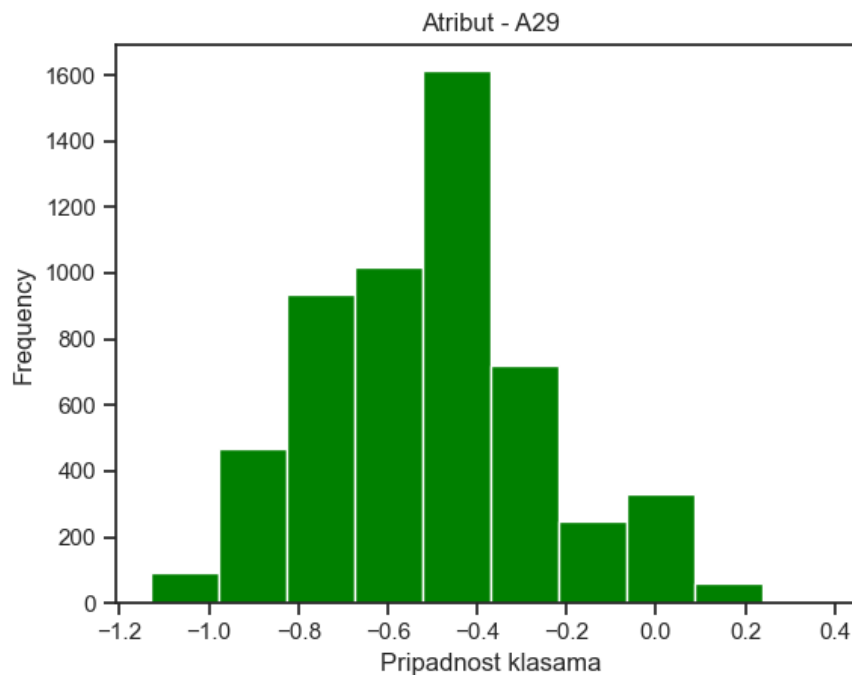
Slika 30 – Grafički prikaz atributa A27

- Atribut br. 28 - A28
  - Opseg vrednosti [-1.154, 0.422]



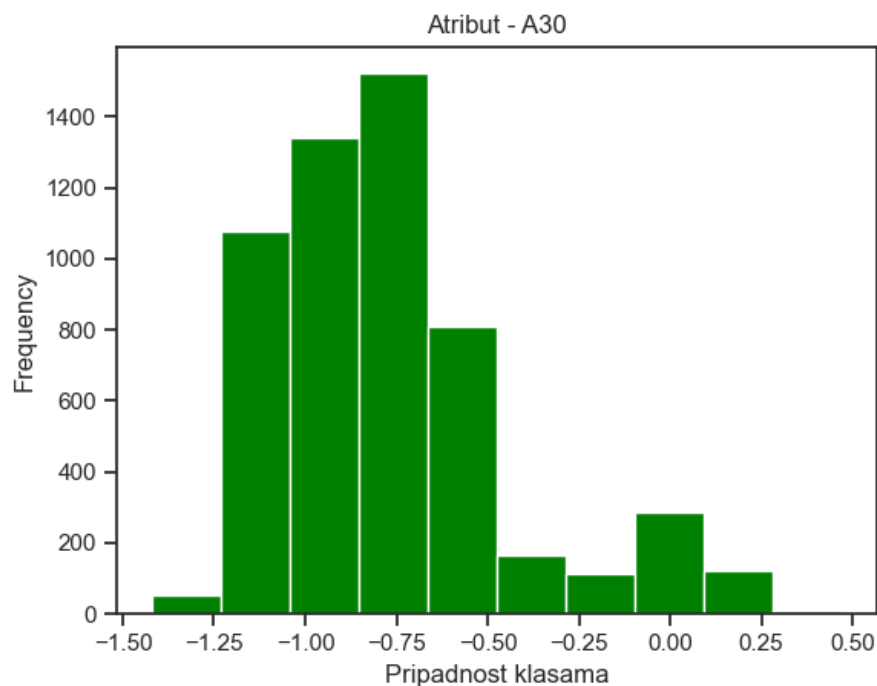
Slika 31 – Grafički prikaz atributa A28

- Atribut br. 29 - A29
  - Opseg vrednosti [-1.132, 0.392]



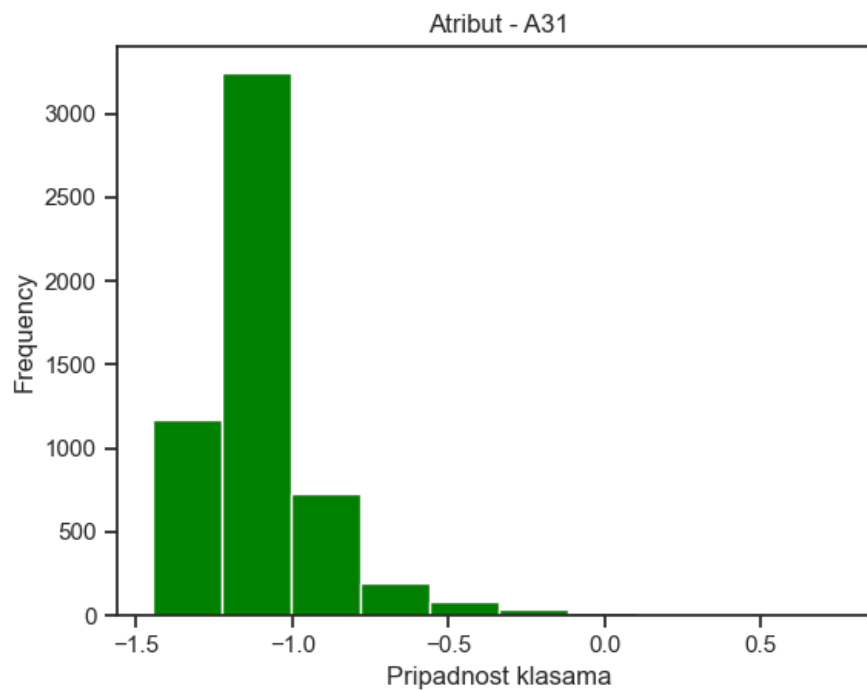
Slika 32 – Grafički prikaz atributa A29

- Atribut br. 30 – A30
  - Opseg vrednosti [-1.422, 0.472]



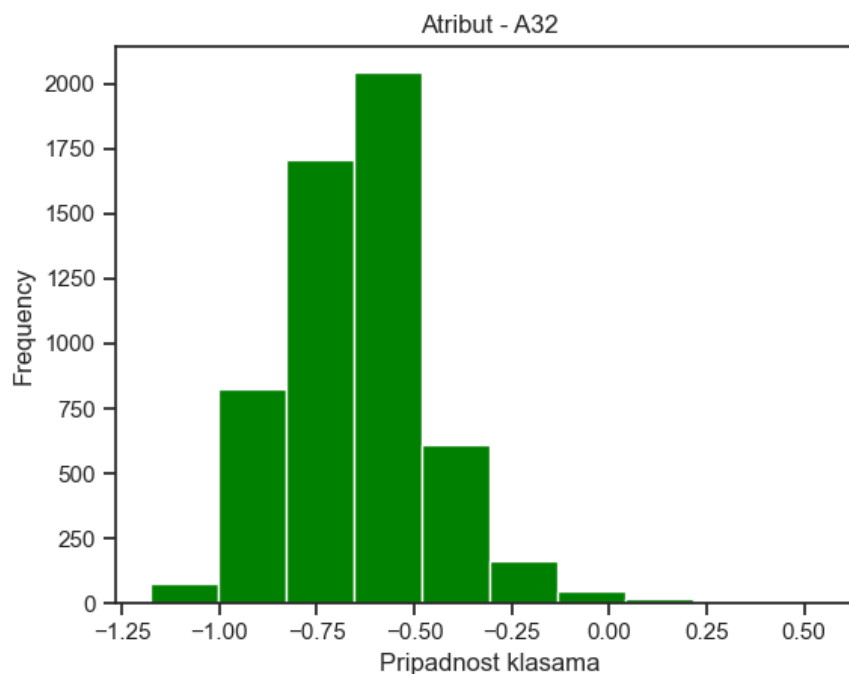
Slika 33 – Grafički prikaz atributa A30

- Atribut br. 31 – A31
  - Opseg vrednosti [-1.45, 0.774]



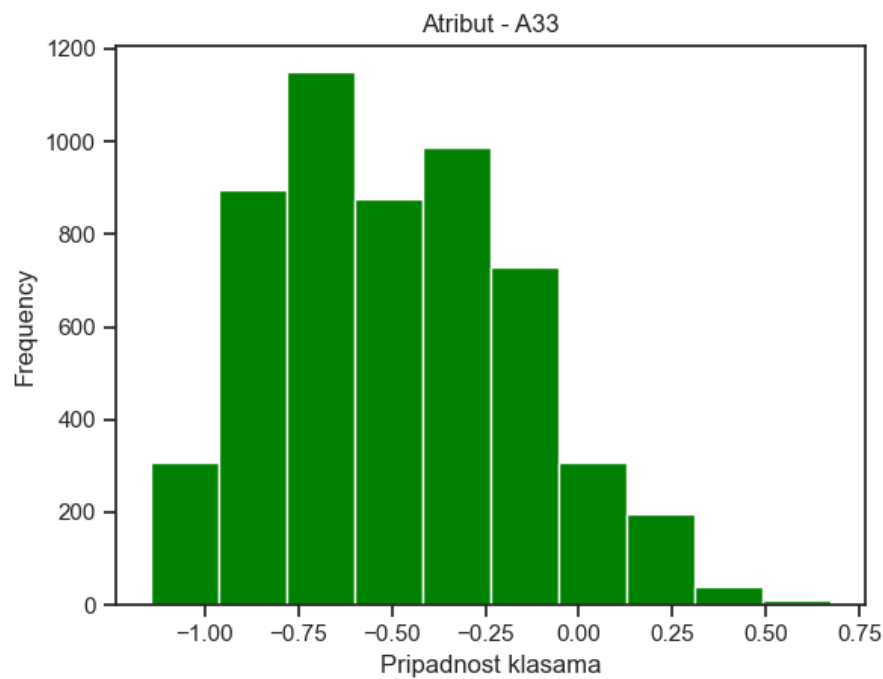
Slika 34 – Grafički prikaz atributa A31

- Atribut br. 32 – A32
  - Opseg vrednosti [-1.179, 0.565]



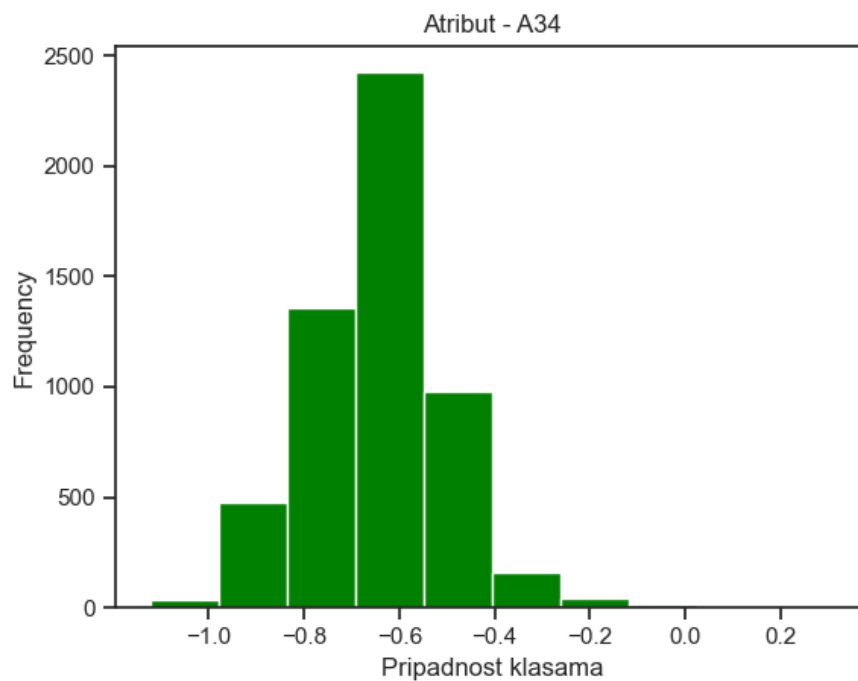
Slika 35 – Grafički prikaz atributa A32

- Atribut br. 33 – A33
  - Opseg vrednosti [-1.147, 0.675]



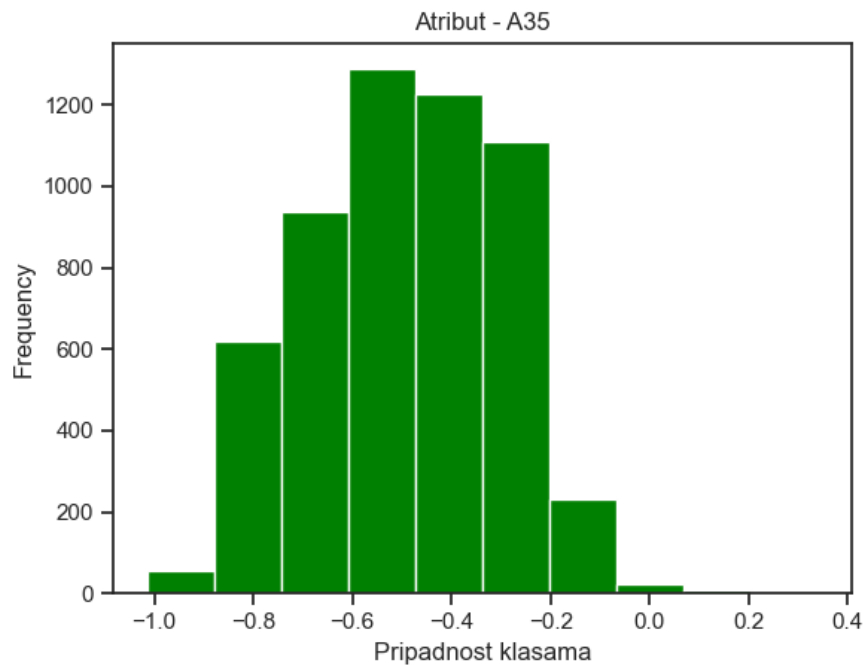
Slika 36 – Grafički prikaz atributa A33

- Atribut br. 34 – A34
  - Opseg vrednosti [-1.123, 0.313]



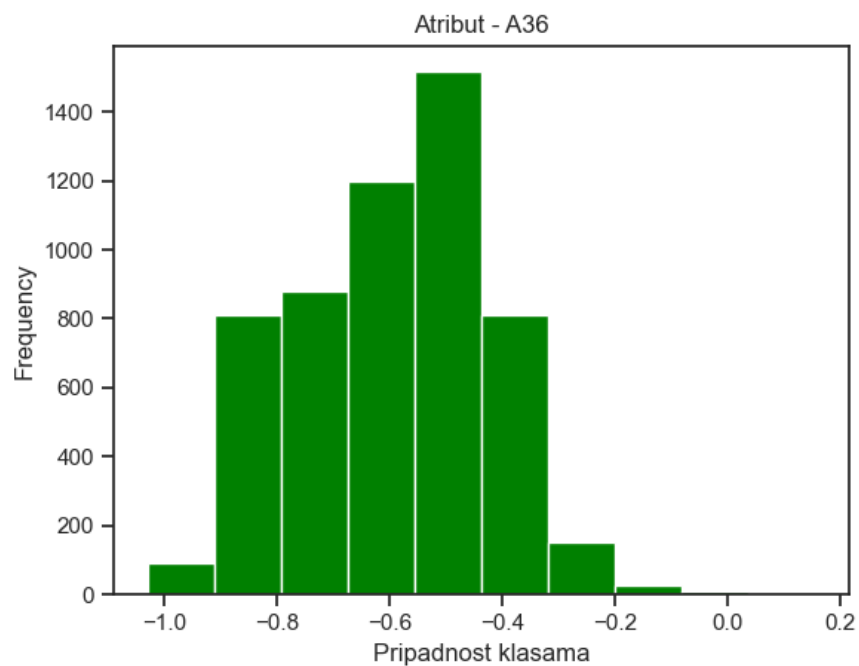
Slika 37 – Grafički prikaz atributa A34

- Atribut br. 35 – A35
  - Opseg vrednosti [-1.015, 0.34]



Slika 38 – Grafički prikaz atributa A35

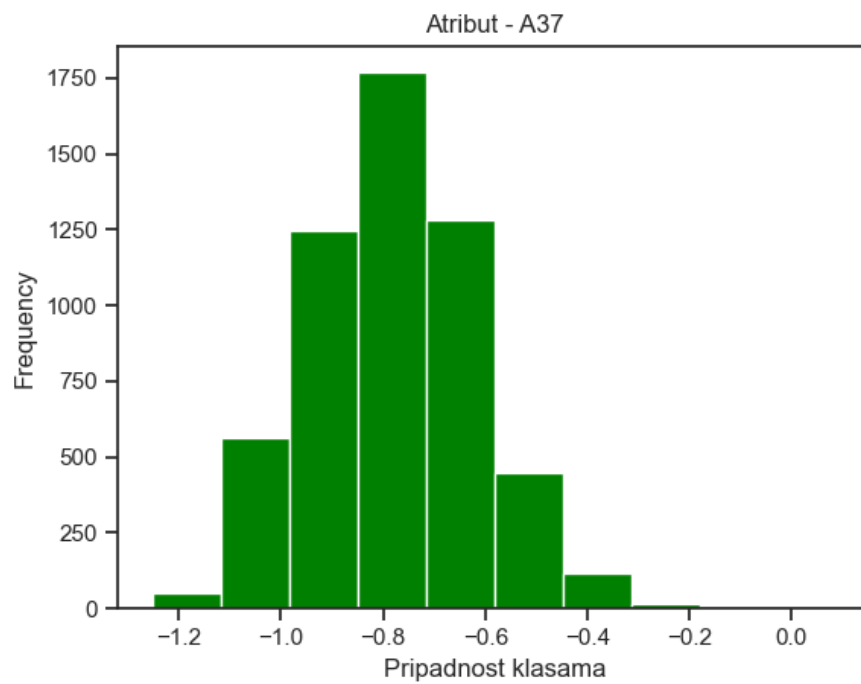
- Atribut br. 36 – A36
  - Opseg vrednosti [-1.03, 0.156]



Slika 39 – Grafički prikaz atributa A36

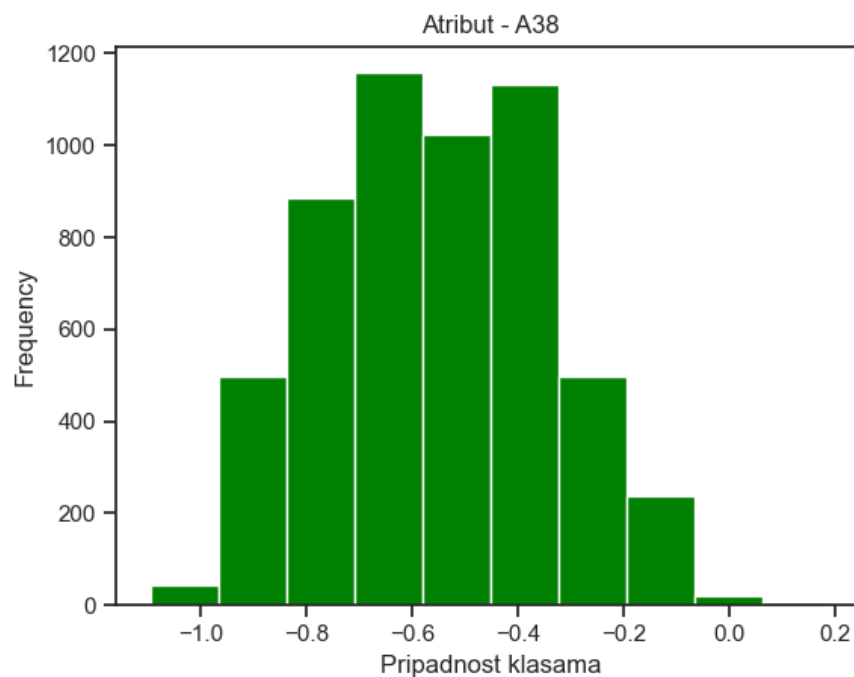


- Atribut br. 37 – A37
  - Opseg vrednosti [-1.253, 0.09]



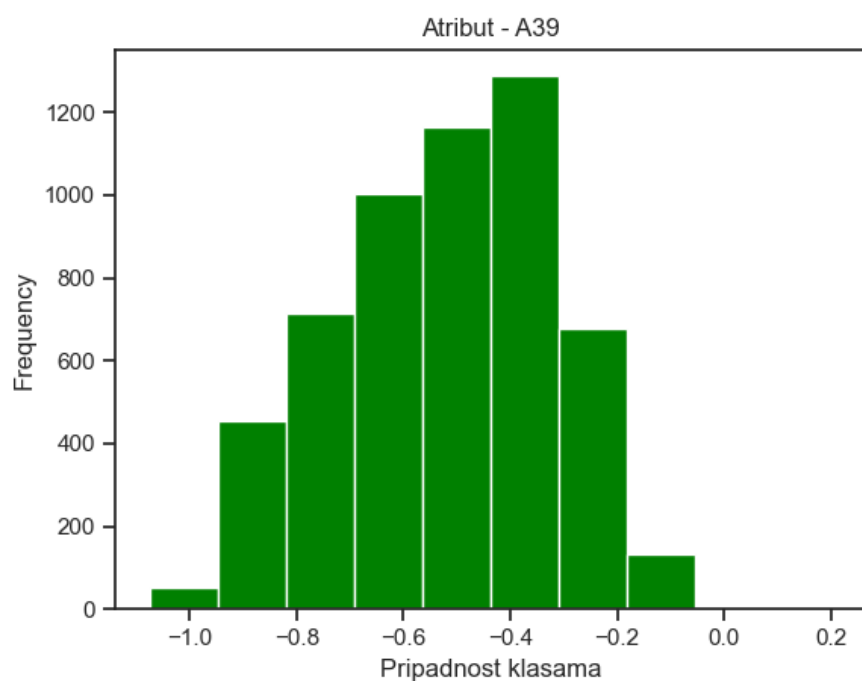
Slika 40 – Grafički prikaz atributa A37

- Atribut br. 38 – A38
  - Opseg vrednosti [-1.097, 0.194]



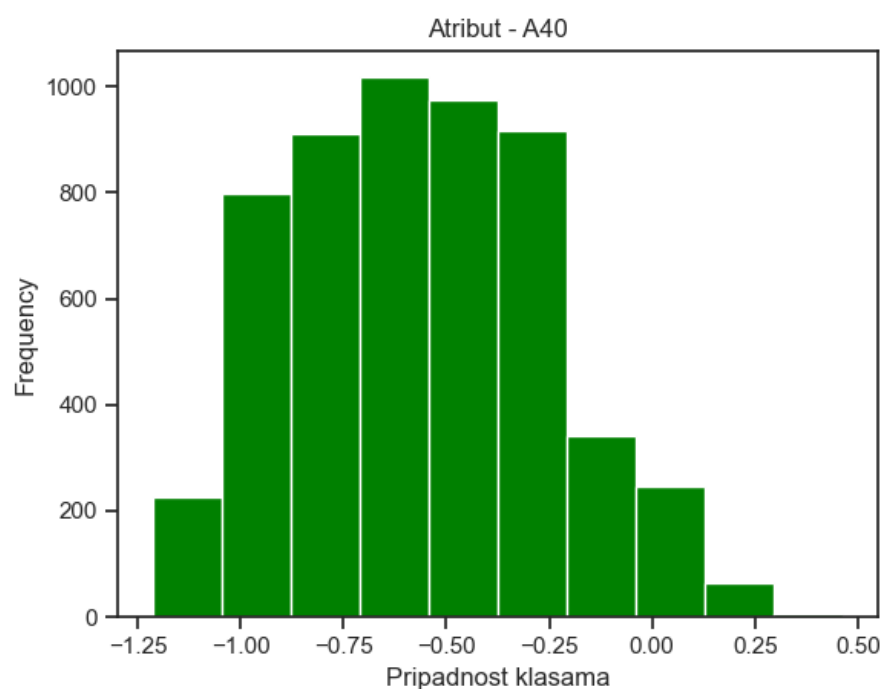
Slika 41 – Grafički prikaz atributa A38

- Atribut br. 39 – A39
  - Opseg vrednosti [-1.076, 0.202]



Slika 42 – Grafički prikaz atributa A39

- Atribut br. 40 – A40
  - Opseg vrednosti [-1.215, 0.465]



Slika 43 – Grafički prikaz atributa A40

Pre samog procesa treniranja mreže, potrebno je podatke podeliti na podatke za treniranje i podatke za testiranje. Zatim odrediti ulazne i izlazne podatke za treniranje neuronske mreže.

Tok pripremanja podataka za obradu dat je u priloženim kodovima.

```
""" Odredjivanje ulaznih i izlaznih podataka - Atributi:Klase """
y = data["class"].values
x = data.drop(["class"], axis=1).values
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=42, test_size=0.2)
```

Slika 44 – Grafički prikaz pripreme podataka

Nakon pripreme podataka, određujemo preciznosti testova, u konkretnom slučaju, testirali smo preko Logističke regresije i Klasifikatora slučajnih šuma, gde smo kao rezultate dobili procentualne vrednosti preciznosti samih testova.

```
""" Logistic Regression """
lr = LogisticRegression(solver="lbfgs")
lr.fit(x_train, y_train)
print("Logistic Regression Classification preciznost testa: {}".format(round(lr.score(x_test, y_test)*100, 2)))
print('\n')

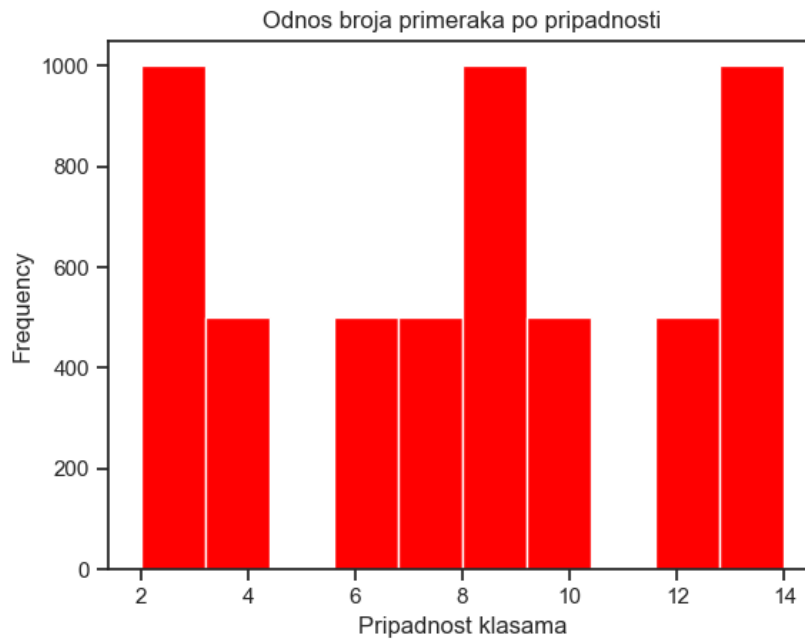
""" Random Forest Classifier """
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(x_train, y_train)
print("Random Forest Classifier preciznost testa: {}".format(round(rf.score(x_test, y_test)*100, 2)))
```

Slika 45 – Grafički prikaz koda za određivanje preciznosti

Logistic Regression Classification preciznost testa: 98.09%

Random Forest Classifier preciznost testa: 97.82%

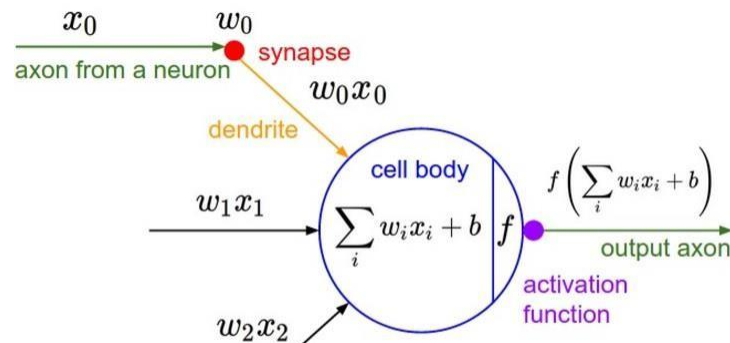
Slika 46 – Grafički prikaz rezultata testova preciznosti



Slika 47 – Grafički prikaz odnosa broja jedinki i pripadnosti klasama

### 3. Neuronska mreža

Na samom početku je navedeno da se ovakav problem klasifikacije rešio primenom neuronskih mreža. Takav pristup je odabran najpre zbog jednostavnosti problema (rezultat je pripadnost određenoj klasi) kao i zbog jednostavnosti prilikom korišćenja neuronske mreže. U nastavku de detaljno biti opisan realizovani pristup kao i njegovi rezultati.



Slika 48 – Blok šema neurona

### 3.1 Priprema radnog okruženja

Realizacija projektnog zadatka je urađena korišćenjem *PyCharm*<sup>(3)</sup> integrisanog razvojnog okruženja kao i *Python 3.9*<sup>(5)</sup> programskog jezika.

Prvo je potrebno instalirati *Python 3.9* i odmah nakon toga *PyCharm* razvojno okruženje, naravno oba 'programa' su besplatna i moguće ih je preuzeti sa oficijelnih sajtova. Nakon završenih instalacija potrebno je instalirati i nekoliko dodataka unutar Python-a koji se inicijalno ne nalaze u njegovom paketu. Reč je o sledećim paketima: *numpy*<sup>(6)</sup>, *pandas*<sup>(2)</sup>, *seaborn*<sup>(8)</sup>, *matplotlib*<sup>(9)</sup>, *sklearn*<sup>(10)</sup>, *keras*<sup>(11)</sup> i *tensorflow*.<sup>(12)</sup>

### 3.2 Kreiranje neuronske mreže

Za kreiranje neuronske mreže koristimo sekvencijalni (*Sequential*) tip mreže koji se nalazi unutar paketa *keras* i *tensorflow*. Takav model predstavlja neuronsku mrežu sa slojevima koji su linearno složeni. Ovakvom modelu je potrebno odmah na početku proslediti broj ulaznih parametara – u ovom slučaju 40, to važi samo za prvi sloj, dok ostali slojevi imaju mogućnost automatske promene oblika tako da nema potrebe za prosleđivanjem ovog parametra i njima. <sup>(4)</sup>

Slojevi se definišu sledećim izvornim kodom:

```
""" Treniranje neuronske mreže """
print("\n Neuronska mreza kroz epohe: \n")

x_train, x_val, y_train, y_val = train_test_split(x_train, y_train, test_size=0.25, random_state=42)

scaler = MinMaxScaler(feature_range=(0, 1))
rescaledx_train = scaler.fit_transform(x_train)
rescaledx_test = scaler.fit_transform(x_test)
rescaledx_val = scaler.fit_transform(x_val)

x_train_normal = rescaledx_train
x_val_normal = rescaledx_val

tf.random.set_seed(0)
model = tf.keras.models.Sequential(layers=[tf.keras.layers.Dense(40, activation="relu"),
                                           tf.keras.layers.Dense(100, activation='relu'),
                                           tf.keras.layers.Dense(15, activation="softmax")])

model.compile(loss = tf.keras.losses.SparseCategoricalCrossentropy(), optimizer = tf.keras.optimizers.SGD(0.1), metrics=["accuracy"])
proces = model.fit(x_train_normal, y_train, epochs=20, validation_data=(x_val_normal, y_val))
print(model.summary())
```

Slika 49 – Izvorni kod kreiranja slojeva neuronske mreže

Metoda dodaje odabrani tip sloja (*Dense*) sekvencijalnom modelu neuronske mreže. *Dense* predstavlja regularni (klasični) sloj povezivanja u neuronskim mrežama.

Proces normalizacije povećava performance i preciznost samog modela. Ovaj proces je izvršen pomoću skaliranja zbog čega je uvezena `MinMaxScaler` funkcija iz `sklearn` biblioteke, dok se za aktivacione funkcije koriste `relu` (*Rectified Linear Unit*) i `softmax`. "Relu" aktivaciona funkcija je najčešće korišćena funkcija kod neuronskih mreža i mašinskog učenja. Ova funkcija radi po principu da vraća 0 ako primi bilo koji negativan ulaz, dok za bilo koju pozitivnu vrednost vraća tu datu vrednost. Iako je jako jednostavna, ova funkcija daje veoma dobre rezultate. "Softmax" je funkcija koja pretvara vektor ulaznih vrednosti u vektor vrednosti čiji je zbir jednak 1. Na kraju, nakon definisanja svih gore navedenih parametara, je potrebno odraditi i kompajliranje mreže gde se kao ulazni argumenti prosleđuju sledeći parametri: `optimizer` (funkcija pomoću koje se radi optimizacija, u ovom slučaju *SGD - Stochastic gradient descent*), `loss` (funkcija gubitaka ili funkcija cilja) i `metrics` (funkcija za ocenu performansi modela).

### 3.3 Treniranje i testiranje neuronske mreže

Da bi se dobili rezultati, potrebno je prvo istrenirati (obučiti - *fit*) mrežu sa trening podacima. Ulazni argumenti su ulazni podaci (ulazni i izlazni podaci), maksimalan broj epoha treninga (obuke) kao i uslov zaustavljanja obučavanja mreže. Maksimalni broj epoha je postavljen na 20 epoha. Nakon toga sledi i testiranje mreže sa test podacima i na osnovu toga se mogu odrediti rezultati, zatim se poređenjem dobijenih rezultata može zaključiti da li je neuronska mreža dobra i da li odrađuje svoj posao onako kako treba.<sup>(20)</sup>

```
model.compile(loss = tf.keras.losses.SparseCategoricalCrossentropy(), optimizer = tf.keras.optimizers.SGD(0.1), metrics=["accuracy"])
proces = model.fit(x_train_normal, y_train, epochs=20, validation_data=(x_val_normal, y_val))
print(model.summary())
```

Slika 50 – Izvorni kod treniranja i testiranja mreže kao i uslova zaustavljanja

### 3.4 Rezultati neuronske mreže

Grafički prikaz promene tačnosti tokom trajanja (epoha) treniranja mreže:

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 40)	1640
dense_1 (Dense)	(None, 100)	4100
dense_2 (Dense)	(None, 15)	1515
Total params: 7,255		
Trainable params: 7,255		
Non-trainable params: 0		
None		

Slika 51 – Rezultat treniranja mreže

Vrednosti prikazane unutar konfuzione matrice nam govore koliko je naša mreža tačno odredila pripadnost jedinki nad test podacima.

U pitanju su dve konfuzione matrice

1. Konfuziona matrica za logističku regresiju
2. Konfuziona matrica za klasifikator slučajnih šuma

Prikaz vrednosti konfuzionih matrica:

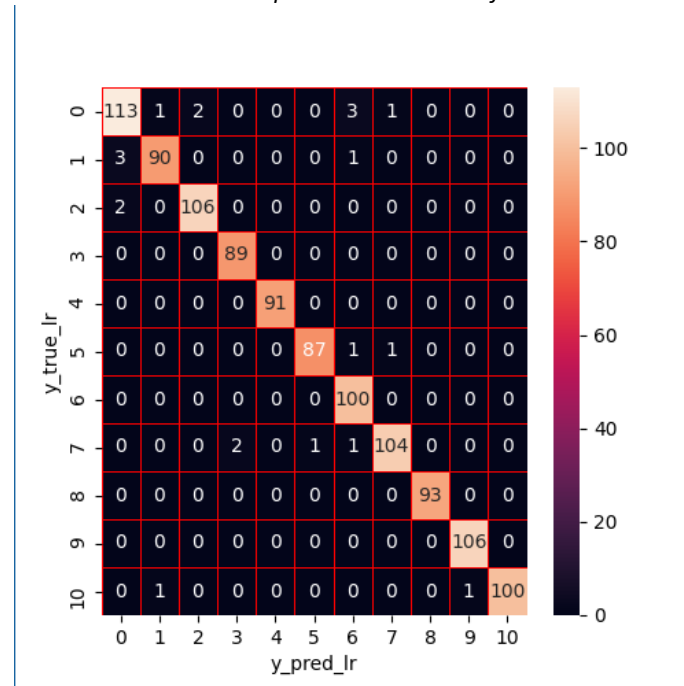
```

""" Crtanje konfuzionih matrica """
""" lr """
y_pred_lr = lr.predict(x_test)
y_true_lr = y_test
cm = confusion_matrix(y_true_lr, y_pred_lr)
f, ax = plt.subplots(figsize=(5, 5))
sns.heatmap(cm, annot = True, linewidths=0.5, linecolor="red", fmt = ".0f", ax=ax)
plt.xlabel("y_pred_lr")
plt.ylabel("y_true_lr")
plt.show()

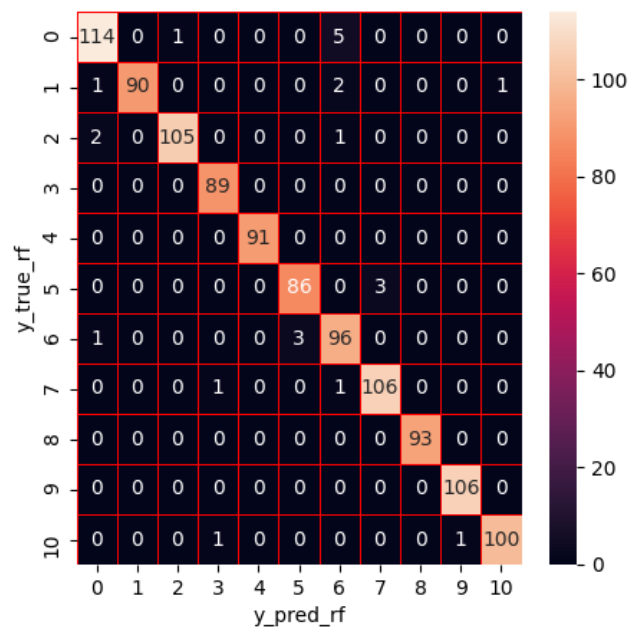
""" rf """
y_pred_rf = rf.predict(x_test)
y_true_rf = y_test
cm = confusion_matrix(y_true_rf, y_pred_rf)
f, ax = plt.subplots(figsize=(5,5))
sns.heatmap(cm, annot = True, linewidths=0.5, linecolor="red", fmt = ".0f", ax=ax)
plt.xlabel("y_pred_rf")
plt.ylabel("y_true_rf")
plt.show()

```

Slika 52 – Izvorni kod za prikaz vrednosti konfuzionih matrica



Slika 53 – Prikaz vrednosti konfuzione matrice za logičku regresiju



Slika 54 – Prikaz vrednosti konfuzione matrice za  
klasifikator slučajnih šuma

Kao poslednji izlaz programa dobijamo rezultate evaluacije tačnosti nad testiranim podacima. Provera tačnosti realizovana je sledećim kodom.

```
""" Evaluacija podataka """
print("\n")
print("Evaluacija na testiranim podacima")
results = model.evaluate(x_test, y_test, batch_size=128)
print("test loss, test acc:", results)
print("\n")
print("Generisanje predvidjanja na 4 uzorka")
predictions = model.predict(x_test[:4])
print("oblik predvidjanja:", predictions.shape)
```

Slika 55 – Prikaz koda za evaluaciju

Rezultati evaluacije prikazani su na priloženoj slici.

```
Evaluacija
Evaluacija na testiranim podacima
1/9 [==>.....] - ETA: 0s - loss: 4.4976 - accuracy: 0.2812
s/step - loss: 4.9078 - accuracy: 0.2809
test loss, test acc: [4.907810211181641, 0.28090909123420715]

Generisanje predvidjanja na 4 uzorka
oblik predvidjanja: (4, 15)
```

Slika 56 – Prikaz rezultata evaluacije



Jako bitna stavka koda jeste učitavanje podataka, svi podaci su učitani iz .csv fajla uz pomoć biblioteke pandas. Podaci su dalje prosleđeni funkcijama za određivanje kolone koja predstavlja klase i funkcije za određivanje kolona u kojima se nalaze podaci učitanih atributa.

Učitavanje podataka se realizuje sledećim kodom.

```
""" Ucitavanje dataseta """
data = pd.read_csv('texture.csv')

""" Sredjivanje izlaza - klasa """
data['class'].unique()

""" Provera da li ima duplikata unutar dataseta """
tot = len(set(data.index))
last = data.shape[0] - tot
print('Ima {} duplikata.\n'.format(last))

""" Provera da li ima null vrednosti unutar dataseta """
null_count = 0
for val in data.isnull().sum():
    null_count += val
print('Ima {} null vrednosti.\n'.format(null_count))

""" Informacije u vezi podataka """
data.info()
```

Slika 57 – Prikaz učitavanja podataka

Definisana je i funkcija za pronalaženje duplikata, kao i funkcija za proveru da li u podacima postoji podatak koji ima null vrednost. Rezultati funkcija su prikazani u priloženim slikama.

Ima 0 duplikata.

Ima 0 null vrednosti.

Slika 58 – Prikaz rezultata funkcija za proveru vrednosti

```

      Broj  Promenljive  RangeIndex: 5500 entries, 0 to 5499
class      11          88  Data columns (total 41 columns):
A1         861          88  #   Column  Non-Null Count  Dtype
A2         979          88  ---  ---
A3        1199          88  0   class    5500 non-null    int64
A4        1072          88  1   A1       5500 non-null    float64
A5        1025          88  2   A2       5500 non-null    float64
A6         961          88  3   A3       5500 non-null    float64
A7         965          88  4   A4       5500 non-null    float64
A8        1003          88  5   A5       5500 non-null    float64
A9        1032          88  6   A6       5500 non-null    float64
A10       1234          88  7   A7       5500 non-null    float64
A11        861          88  8   A8       5500 non-null    float64
A12        894          88  9   A9       5500 non-null    float64
A13       1300          88  10  A10      5500 non-null    float64
A14        696          88  11  A11      5500 non-null    float64
A15        810          88  12  A12      5500 non-null    float64
A16        727          88  13  A13      5500 non-null    float64
A17        805          88  14  A14      5500 non-null    float64
A18        899          88  15  A15      5500 non-null    float64
A19        852          88  16  A16      5500 non-null    float64
A20       1282          88  17  A17      5500 non-null    float64
A21        861          88  18  A18      5500 non-null    float64
A22        990          88  19  A19      5500 non-null    float64
A23       1223          88  20  A20      5500 non-null    float64
A24       1150          88  21  A21      5500 non-null    float64
A25       1112          88  22  A22      5500 non-null    float64
A26       1100          88  23  A23      5500 non-null    float64
A27       1010          88  24  A24      5500 non-null    float64
A28       1082          88  25  A25      5500 non-null    float64
A29       1110          88  26  A26      5500 non-null    float64
A30       1217          88  27  A27      5500 non-null    float64
A31        861          88  28  A28      5500 non-null    float64
A32        898          88  29  A29      5500 non-null    float64
A33       1309          88  30  A30      5500 non-null    float64
A34        742          88  31  A31      5500 non-null    float64
A35        838          88  32  A32      5500 non-null    float64
A36        750          88  33  A33      5500 non-null    float64
A37        797          88  34  A34      5500 non-null    float64
A38        921          88  35  A35      5500 non-null    float64
A39        865          88  36  A36      5500 non-null    float64
A40       1252          88  37  A37      5500 non-null    float64
          88  38  A38      5500 non-null    float64
          88  39  A39      5500 non-null    float64
          88  40  A40      5500 non-null    float64

```

Slika 59 – Prikaz rezultata funkcija za učitavanje podataka

## **4. Zaključak**

Na osnovu svih vrednosti parametara - rezultata, može se zaključiti da je za navedeni problem klasifikacije ovaj algoritam (neuronske mreže) veštačke inteligencije veoma zadovoljavajuće odradio svoj posao.

Algoritam je bez ikakvih grešaka i za kratko vreme u potpunosti uradio tačnu klasifikaciju i može se zaključiti da je odabir ovog algoritma za opisani problem odličan.

## **Literatura**

### **Web:**

- <sup>1</sup> Univerzitet u Beogradu – Neuronske mreže. Preuzeto: Maj 15, 2021 - [https://www.mas.bg.ac.rs/\\_media/biblioteka/izdanja/1/1.00001.pdf](https://www.mas.bg.ac.rs/_media/biblioteka/izdanja/1/1.00001.pdf)
  - <sup>2</sup> KEEL. Preuzeto: April 23, 2021 - <https://sci2s.ugr.es/keel/dataset.php?cod=72>
  - <sup>3</sup> Stackoverflow. Preuzeto: Maj 15, 2021 - <https://stackoverflow.com>
  - <sup>4</sup> PyCharm. Preuzeto: Januar 17, 2021 - <https://www.eclipse.org>
  - <sup>5</sup> Python. Preuzeto: Januar 17, 2021 - <https://www.python.org>
  - <sup>6</sup> NumPy. Preuzeto: April 11, 2021 - <https://www.numpy.org>
  - <sup>7</sup> Pandas. Preuzeto: April 11, 2021 - <https://pandas.pydata.org>
  - <sup>8</sup> Seaborn. Preuzeto: April 11, 2021 - <https://seaborn.pydata.org>
  - <sup>9</sup> Matplotlib. Preuzeto: April 11, 2021 - <https://matplotlib.org>
  - <sup>10</sup> Sklearn. Preuzeto: April 12, 2021 - <https://scikit-learn.org/stable>
  - <sup>11</sup> Keras. Preuzeto: April 12, 2021 - <https://keras.io>
  - <sup>12</sup> Tensorflow. Preuzeto: April 12, 2021 - <https://www.tensorflow.org/install>
-