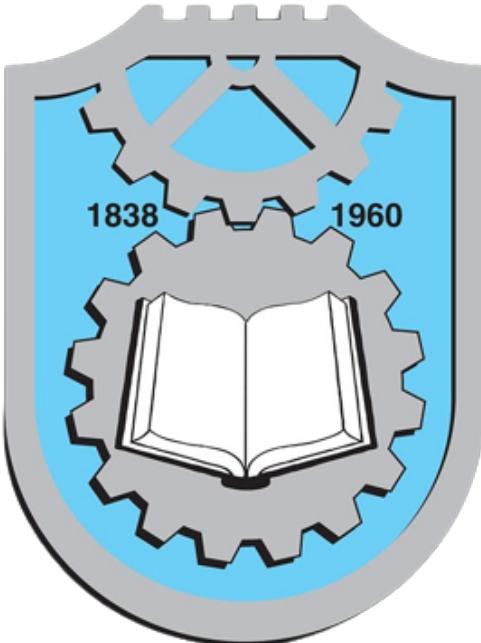


Универзитет у Крагујевцу  
Факултет инжењерских наука



## Вештачка интелигенција

Семинарски рад

### Класификација возила на основу силуете

Професор:

Проф. др. Весна Ранковић

Студент:

Каришић Ђорђе 657/2019

— . — . 2023.

## Seminarski rad iz predmeta Veštačka inteligencija

Data je postavka problema i na:

<https://sci2s.ugr.es/keel/dataset.php?cod=68>

Predstavljeni problem treba rešiti primenom neuronskih mreža, genetskim algoritmom ili nekom od kombinacija algoritama veštačke inteligencije. Kao programski jezik se preporučuju Java, C++ i sl. Matlab se može isključivo koristiti za brzu proveru izabranog algoritma i kao verifikacija dobijenih rezultata.

U samom seminarskom radu obaveno je:

- Objasniti zadati problem, ulazne i izlazne podatke
- Vizualizovati podatke
- Objasniti korišćene algoritme i opravdati njihovo korišćenje
- Diskutovati rezultate i izvesti zaključke
- Objasniti prednosti i mane korišćene metode

Seminarski rad je potrebno poslati na [tijanas@kg.ac.rs](mailto:tijanas@kg.ac.rs) (izvorni kod programa, izvršna verzija, pomoćne datoteke itd.) i predati u pisanom obliku najkasnije do **28.05.2023.**

Pisani deo seminarskog rada povezan u spiralu treba da sadrži sledeće delove:

- naslovnu stranu sa jasno napisanim osnovnim podacima
- postavku zadatka
- opis delova programa sa ilustracijama i samim izvornim kodom
- spisak korišćene literature

Kandidat je u obavezi da prilikom odbrane praktičnom demonstracijom rada programa detaljno objasni rad pojedinih delova programa.

U Kragujevcu, 28.03.2023. godine

# Садржај

<b>1 Увод</b>	<b>3</b>
<b>2 Анализа скупа података</b>	<b>4</b>
<b>3 Модели и алгоритми у оквиру система</b>	<b>7</b>
3.1 Неуронска мрежа, приступ дубоким учењем . . . . .	7
3.2 Приступ другим алгоритмима машинског учења . . . . .	11
3.2.1 Стабло одлучивања . . . . .	11
3.2.2 К најближих суседа . . . . .	12
3.2.3 Метода потпорних вектора . . . . .	12
<b>4 Резултат класификације и могућа побољшања</b>	<b>13</b>
4.1 Резултујућа прецизност употребљених алгоритама . . . . .	13
4.2 Потенцијални проблеми и предлог решења . . . . .	13

# 1 Увод

Проблем дефинисан скупом података предлаже коришћење вештачких неуронских мрежа и различитих алгоритама машинског учења за корист класификације и детекције возила на основу карактеристика сиљуете истог. Постоје четири предефинисане класе, и то су:

1. Комби
2. Saab
3. Аутобус
4. Opel

На основу информација попут дужине, ширине или компактности сенке/сиљуете потребно је јасно класификовати податак у једну од претходно наведених класа. Један тип или класа возила може бити представљен кроз више различитих углова, што чини сенку другачијом. Подаци су представљени **.dat** форматом и захтевају трансформисање у други формат како би мрежа адекватно обрадила сваки податак. Податке је неопходно анализирати и визуелизовати.

Предлог алата за синтезу потребног система јесте програмски језик **Python**. Python библиотеке за визуелизацију и анализу података:

- **matplotlib**
- **seaborn**
- **visualkeras**

Python библиотеке за синтезу система:

- **pandas**
- **keras**
- **kerastuner**
- **numpy**
- **sklearn**

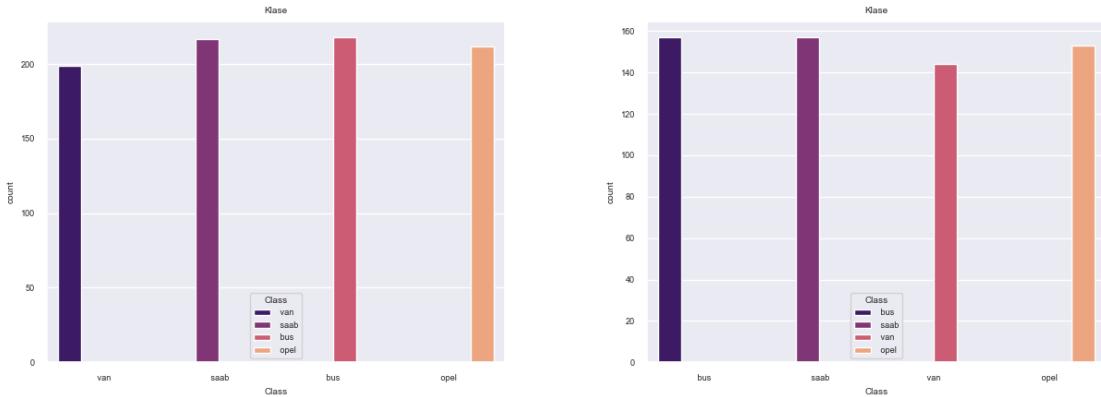
Уз помоћ наведених библиотека могуће је направити систем класификације неопходан за решавање задатог проблема.

## 2 Анализа скупа података

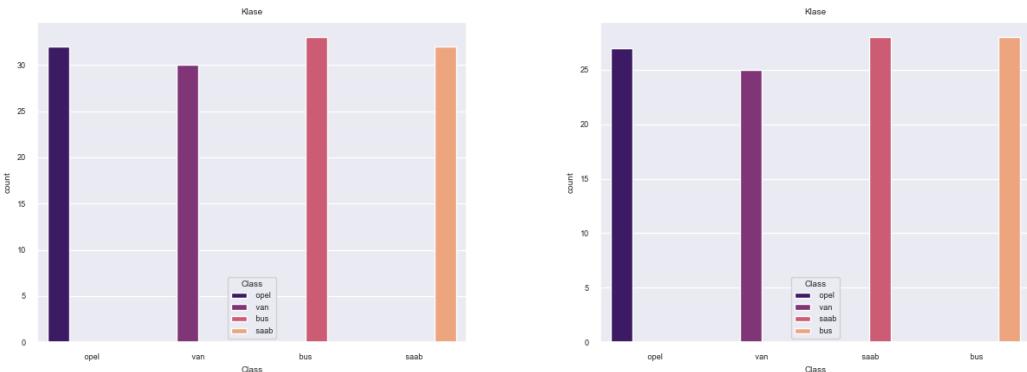
Скуп података садржи 862 редова података чије карактеристике су распоређене у 19 колона. Последња колона говори о типу возила.

Потребно је проверити интегритет података тако што је скуп потребно адекватно претражити за неодговарајуће формате података (Null, NaN,...).

Дати скуп података је потребно поделити на тренинг, валидациони и тест скуп, примењена размера ових скупова биће 70%:15%:15% респективно. Сликама 1 и 2 приказани су графици који указују на распоред класа по скуповима података.



Слика 1: Дистрибуција класа на агрегираном и тренинг скупу података, респективно

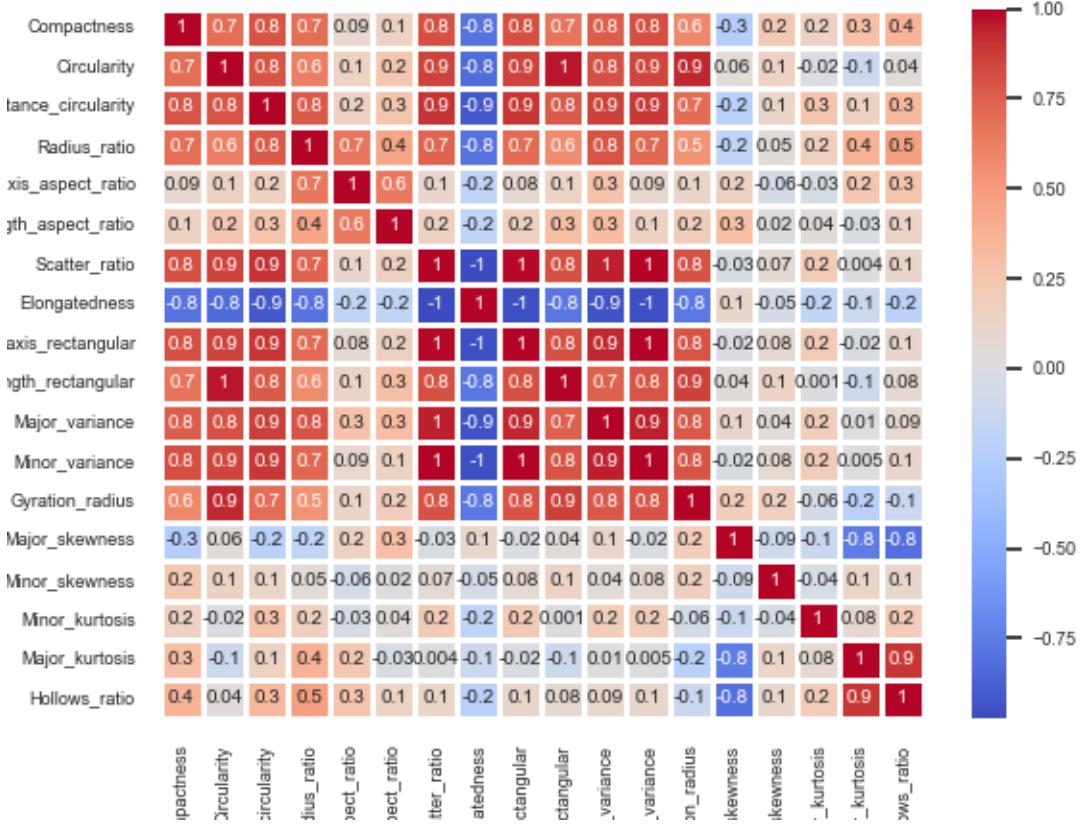


Слика 2: Дистрибуција класа на тест и валидационом скупу података, респективно

Податке је неопходно поделити тако да распоред класа у подскуповима буде сличне дистрибуције како би тестирање и валидација мреже били успешни. На основу слика 1 и 2 може се закључити да је то испуњено.

Због природе проблема, потребно је извршити кодирање података који нису нумерички. У оквиру практичног дела задатка урађено је one-hot кодирање.

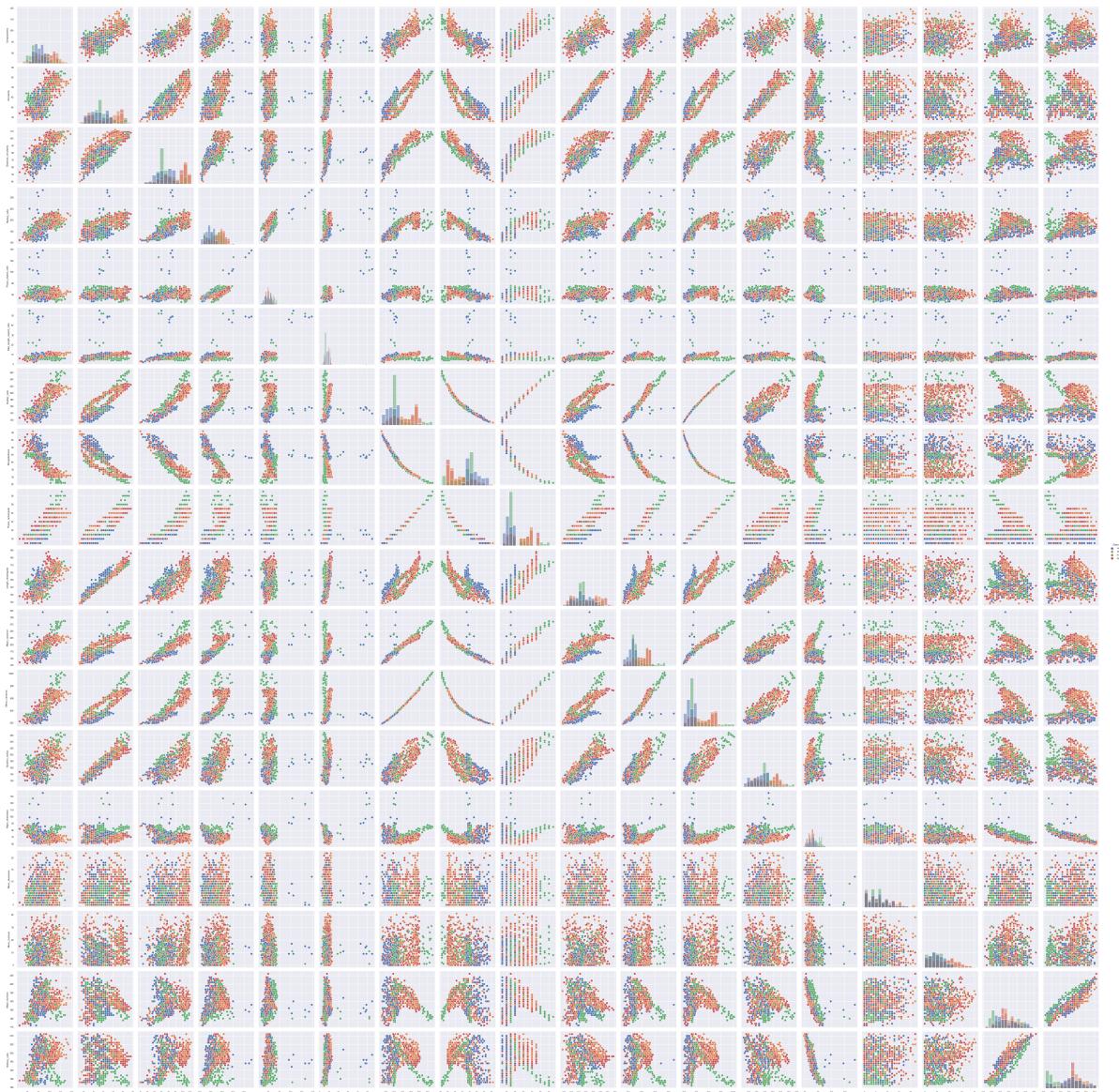
Како би тачност система била задовољавајућа, потребно је проверити везе између карактеристика података и утврдити да нема било каквих аномалија, попут апсурдно велике међусобне зависности. На слици 3 приказана је корелациона матрица која говори о везама између карактеристика података.



Слика 3: Корелациона матрица

На основу слике, може се закључити да је већи број карактеристика узајамно зависан од других. Ова чињеница намеће претпоставку да систем неће бити стабилан уколико се задрже карактеристике које зависе од других, јер ће модел теже успоставити зависност једне издвојене карактеристике у односу на карактеристику која се посматра као жељени излаз система, ако њена вредност зависи од других карактеристика.

График на слици 4 приказује податке распоређене по класама у равни помоћу X и Y осе које су представљене помоћу распона вредности две издвојене карактеристике, за сваку комбинацију карактеристика. Овај тип графика назива се **Pair grid** график, и омогућава визуелну представу веза између карактеристика и класа. Свака обојена тачка представља један податак, који у зависности од боје припада једној класи. Како постоји 18 карактеристика сваког податка (изузев класе), генерише се матрица графика  $18 \times 18$  како би се визуелизовала веза између сваке две карактеристике.



Слика 4: Pairgrid матрица графика

### 3 Модели и алгоритми у оквиру система

#### 3.1 Неуронска мрежа, приступ дубоким учењем

Дефинисање система који за проблем вишекласне класификације података даје модел вештачке неуронске мреже налаже решавање проблема оптимизације архитектуре и хиперпараметара таквог модела.

Како би се пронашао адекватан модел неуронске мреже за решавање дефинисаног проблема, користи се **keras tuner** библиотека која служи за подешавање хиперпараметара и одређивање њихових перформанси помоћу **Hyperband** методе.

---

```
1 def getModel(hp:kt.Hyperband)->Sequential:
2     model = Sequential()
3     model.add(Input(shape=(18,)))
4     hp_learning_rate = hp.Choice('learning_rate',
5         values=[1e-1,1e-2, 1e-3, 1e-4])
6     for i in range(1, hp.Int("num_layers", 1, 4)):
7         model.add(
8             Dense(
9                 units=hp.Int("units_" + str(i), min_value=2, max_value=192, step=1),
10                activation="relu", kernel_regularizer=regularizers.l2(0.01))
11            )
12    model.add(Dense(4, activation='softmax'))
13    model.compile(loss='categorical_crossentropy',
14        optimizer=keras.optimizers.Adam(learning_rate=hp_learning_rate),
15        metrics=['accuracy'])
16
17    return model
```

---

Код изнад представља функцију која дефинише модел и границе хиперпараметара, на основу којих Hyperband алгоритам тражи оптималан модел.

За границе простора претраживања конфигурација хиперпараметара постављене су вредности из наредне табеле:

Хиперпараметар	Од	До	Остало
Стопа учења	0.0001	0.1	-
Број скривених слојева	1	4	-
Број неурона у слоју	2	192	Корак инкрементације 1

Комплетан процес подешавања хиперпараметара се извршава под константним бројем епоха (100, уколико не дође до прекида), фактором стрпљења (5), прати се прецизност модела над валидационим скупом, под условом да модел прати фактор стрпљења. Модел прекида тренинг након итерације  $i$ , за коју важи:

$$CE(i) \geq CE(i-1) \geq CE(i-2) \geq \dots \geq CE(i-(f_p - 1)),$$

Где је  $CE(i)$  вредност функције губитака (**Categorical cross-entropy**) у  $i$ -тој итерацији тренинга, а  $f_p$  целобројна вредност фактора стрпљења.

---

```

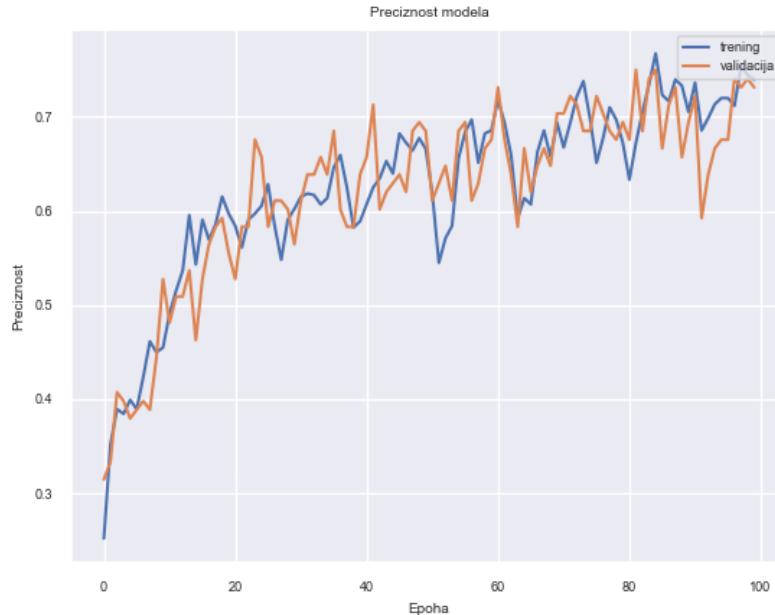
1 tuner = kt.Hyperband(getModel,
2                     objective= 'val_accuracy',max_epochs=100,factor=3,
3                     directory= 'dir', project_name= 'hiperparam_modeli')
4
5 stop_early = tf.keras.callbacks.EarlyStopping(monitor= 'val_loss', patience=10)
6 tuner.search(X_train, y_train, epochs=100,validation_data=(X_val, y_val),
7 callbacks=[stop_early],batch_size=64)
8 all_hps = tuner.get_best_hyperparameters(num_trials=5)
9 best_hp=all_hps[0]
10
11 h_model = tuner.hypermodel.build(best_hp)
12 history=h_model.fit(X_train, y_train, epochs=100, validation_data=(X_val, y_val),
13 batch_size=64)
14 h_model.summary()
15 h_eval_dict = h_model.evaluate(X_test, y_test, return_dict=True)

```

---

Код изнад претражује вишедимензиони простор методом Hyperband и пронађени модел тренира над тренинг скупом података.

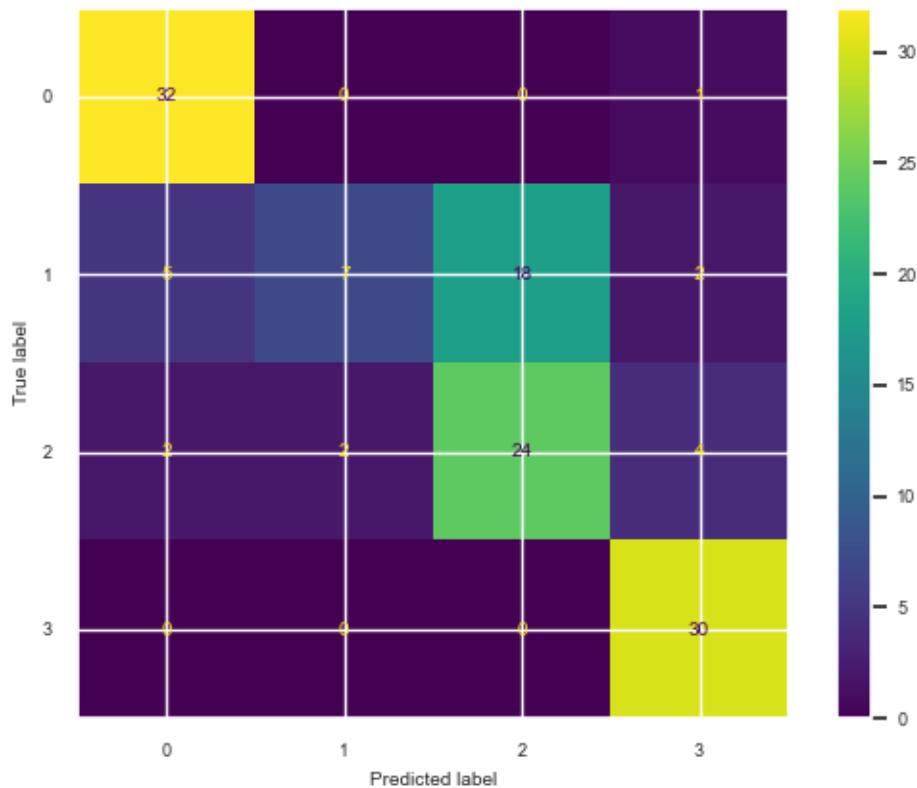
Резултат претраге простора хиперпараметара је пет најбољих конфигурација по прецизношћу над валидационом скупу података, где приоритет имају мање сложене конфигурације. Најбоља конфигурација се међу њима се прослеђује новом моделу који се излаже традиционалном тренирању над истим подацима. Дати модел се евалуира над тестним скупом података, и генерише се извештај о прецизности модела.



Слика 5: Прецизност на валидационом и тренинг скупу

На основу слике 5 може се приметити волатилност прецизности у односу на број епохе. Тада проблем може настати услед мале димензије подскупа тренирања (batch size), велиоког степена учења или неких карактеристика у оквиру скупа података. Обзиром да у току подешавања хиперпараметара је наведено више степена учења, и величина димензије подскупа тренирања није мала, доноси се претпоставка да постоје проблеми у оквиру скупа података. Са графика 3 претпоставка се може потврдити, како је јасно да карактеристике из скупа података зависе доста једне од других, јавља се проблем **мултиколинеарности**.

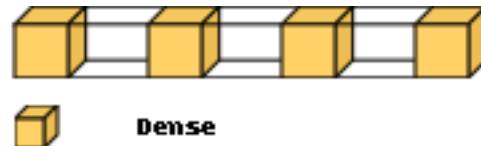
На слици 6 приказана је матрица конфузије.



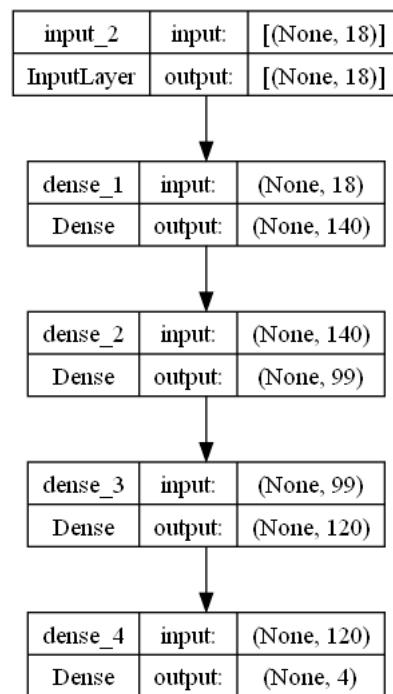
Слика 6: Матрица конфузије

Са матрице конфузије 6 може се закључити да систем има потешкоћа у препознавању разлика између класе 1 и класе 2, конкретно, грешка настаје у току класификације податка који припада класи 1, који модел види као податак из класе 2. Узрок тога је сличност између те две класе, како обе класе представљају веома сличне аутомобиле (Saab и Opel). Такође, на основу способности модела да распознаје класу 0 и класу 3 скоро без грешке може се навести да је модел задовољавајуће способности класификације.

Пронађени модел графички приказан на 7 и 8 је оптимално решење дефинисаног проблема. Постиже тачност од око 73% на сваком скупу података.



Слика 7: Графички приказ модела



Слика 8: Приказ модела по слојевима

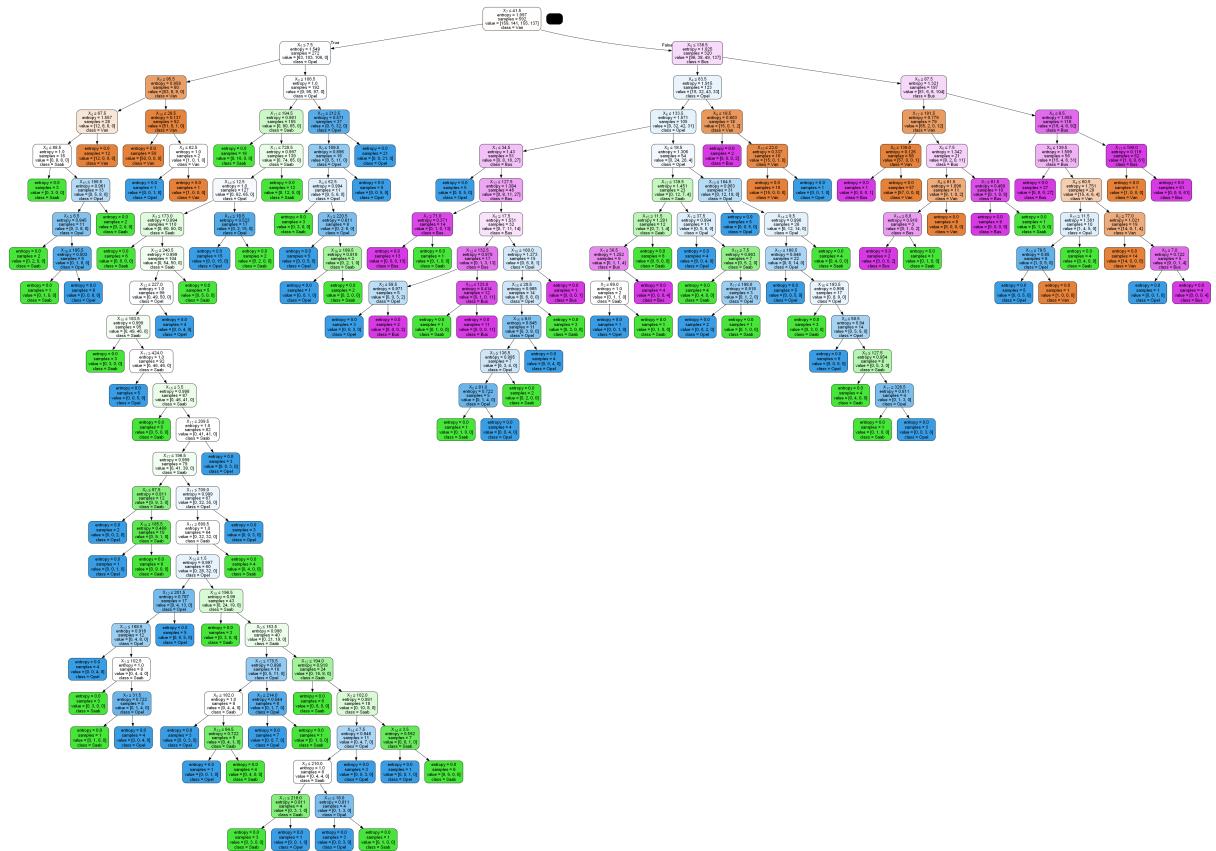
Метода оптимизације враћа 5 најпрецизнијих архитектура модела, са њиховим бројем слојева и стопом учења.

Модел ID	Стопа учења	Број слојева	Тачност
228	0.001	4	73%
144	0.001	4	72%
204	0.01	2	70%
229	0.001	2	69%
237	0.001	4	67%

## 3.2 Приступ другим алгоритмима машинског учења

### 3.2.1 Стабло одлучивања

Стабло одлучивања, са неограниченом дубином и скупом података подељеним размером 70:30 у тренинг и тест скуп, постиже тачност од око 70%. Слика 9 приказује стабло одлучивања конструисано за овај проблем.



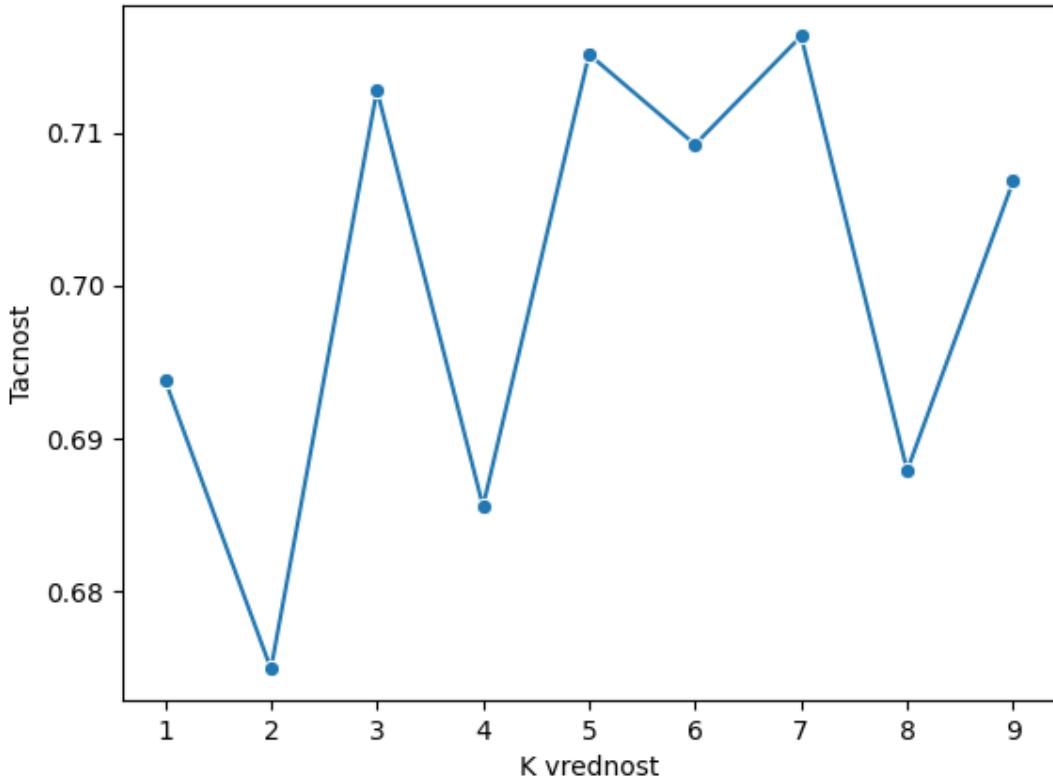
Слика 9: Стабло одлучивања

Тачност овог алгоритма је идентична тачности дубоке неуронске мреже.

На основу дубине и асиметричности стабла, може се приметити аномалија. Примећена аномалија указује на најдубљи део стабла, који за задатак има разликовање класе 1 и класе 2. Ова аномалија указује на тачност претпоставке донесене у оквиру секције 3.1, која предлаже да "систем има потешкоћа у препознавању разлика између класе 1 и класе 2, конкретно, грешка настаје у току класификације податка који припада класи 1, који модел види као податак из класе 2".

### 3.2.2 К најближих суседа

К најближих суседа са скалираним скупом података подељеним размером 70:30 у тренинг и тест скуп, тестирано за вредности  $K \in [1, 2, \dots, 10]$  постиже резултате приказане на слици 10.



Слика 10: К најближих суседа

Најбољи резултат је постигнут са  $K = 7$ , где је тачност мало изнад 70%. Овај резултат говори о томе да су подаци груписани тако да су неке од класа дosta сличне, водећи се тврђњом из секције 3.1, предпоставља се да су подаци који припадају класама 1 и 2 јако сличних карактеристика и тешко их је развојити.

### 3.2.3 Метода потпорних вектора

Метода потпорних вектора са скупом података подељеним 70:30 у тренинг и тест скуп, постиже тачност од 78% са линеарним језгром и 44% са RBF језгром. Овај резултат потврђује претпоставку донесену у процењивању претходних метода.

## 4 Резултат класификације и могућа побољшања

### 4.1 Резултујућа прецизност употребљених алгоритама

Упоређивање резултата коришћених метода класификације возила на основу силуете дато је наредном табелом:

ID	Метода	Тачност	Додатне информације
1	Неуронска мрежа	73%	Подешени хиперпараметри
2	Стабло одлучивања	70%	Без ограничења дубине стабла
3	К најближих суседа	71%	Тестирано за $K \in [1, 2, \dots, 9]$
4	Метода потпорних вектора	78%	Линеарно језгро

Прецизност сваког алгоритма се може дефинисати неједначином  $70\% \leq P_{test} \leq 80\%$  где  $P_{test}$  представља прецизност алгоритма над тестним скупом података. Са табеле се јасно примећује да сваки алгоритам има потешкоћа са класификацијом података, и да имају релативно сличну прецизност.

Најбољи резултат даје **метода потпорних вектора** са линеарним језгром, са тачношћу 78%.

### 4.2 Потенцијални проблеми и предлог решења

У оквиру задатка пронађено је више потенцијалних проблема који отежавају систему да оптимално класификује податке. Предлог решења тих проблема дат је листом:

- **Мултиколинеарност**

Мултиколинеарност је проблем који се јавља када две или више карактеристика зависе једне од других. Модел не може да одреди значај појединачних карактеристика и тежину које оне носе за класификацију. Решење проблема јесте избацување колона које имају веома високу или ниску вредност у корелационој матрици (3).

- **Разликовање идентичних класа**

На слици 6 примећује се да систем скоро па не налази разлику између класа Saab и Opel. Ове класе имају сличне атрибуте, па самим тим, представљају стваран проблем у класификацији. Решење овог проблема јесте груписање ове две класе у једну, нпр. **Cars**. Познато је да је систем способан да реши проблем класификације између класа комби и аутобус готово без грешке, и скоро све податке који припадају или класи Saab или класи Opel класификује у једну од њих, што указује на могуће проблеме у скупу података између ове две класе.

- **Допуна скупа података**

Како скуп података садржи само око 850 редова на 18 улазних карактеристика, може доћи до тога да модел нема доволно информација за налажење комплекснијих веза између атрибута и излаза и самим тим за адекватну класификацију. Сама вештачка допуна скупа података ће донести само вештачке податке, али уколико је алгоритам иза генерирања додатних података конструисан добро, може довести до благих побољшања.