Razpoznavanje vzorcev - implementacija projekta

# *Uporaba vmesnika za namensko programiranje z namenom razpoznavanja vzorcev*

**Avtor: Timotej Petrovčič**
**Mentorja: izr. prof. dr. Simon Dobrišek, as. dr. Klemen Grm**

Univerza v Ljubljani
Fakulteta za elektrotehniko

`Predmet: Razpoznavanje vzorcev`
`Datum: 8 Januar 2023`

---

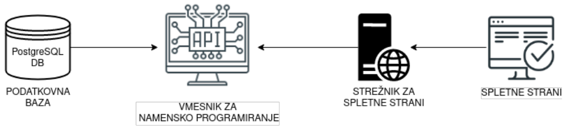
**Kazalo:**

**Ključne besede:** Vmesnik za namensko programiranje (API - Application programming interface), strežnik, odjemalec, podatkovna baza, transformacija Karhunea in Loeveja (PCA - analiza glavnih komponent), razvrščanje z enim najbližjim sosedom (1-NN)

---

## 1 Namen

Namen projekta je izdelava vmesnika za namensko programiranje na področju razpoznavanja vzorcev ter izdelava tako strežniške kot odjemalniške strani (backend/frontend). Prvotno smo se omejili na uvoz podatkov iz "UCI machine learning" repozitorija, saj so vzorčni podatki strukturirani tako, da vsak vzorec predstavlja novo vrstico v poljubni datoteki.



*Slika 1: Struktura delovanja*

## 2 Vmesnik za namensko programiranje

Vmesnik za namensko programiranje (Od sedaj naprej vmesnik) je bil napisan v programskem jeziku Python ter se v grobem deli na dva nivoja in sicer:

- Nivo spletnega dostopa ("recognition_api")
- Nivo podatkovnega dostsopa ("recognition_DAL")

Omenjena ločitev je pomembna, ker loči spletno stran vmesnika od podatkovne strani ter s tem omogoči neodvisno delovanje obeh nivojev, prepreči neželene napake v programu in izboljša varnost celotnega sistema.

Za izdelavo vmesnika smo uporabili asinhrone knjižnice, ki omogočajo sočasno/asinhrono izvajanje operacij (Concurent execution).

### Nivo spletnega dostopa

Nivo spletnega dostopa je v grobem sestavljen iz 4 delov:

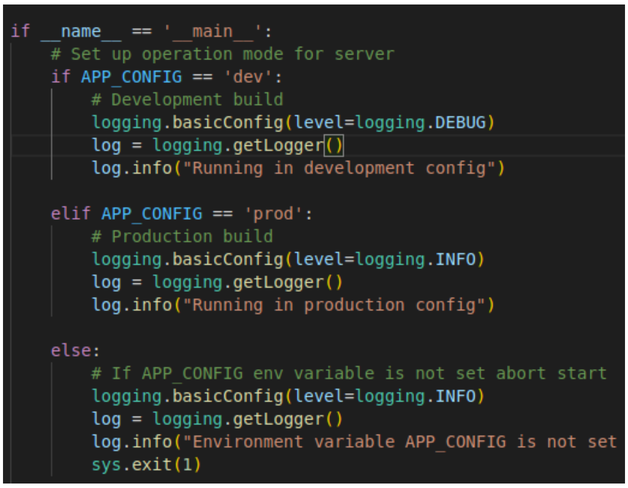
- Uvoz spremenljivk okolja
- Določitev različice izgradnje
- Določitev statičnih poti
- Inicializacija strežniške strani ter zagon vmesnika

Uvoz spremenljivk okolja omogoča dinamično upravljanje s programom preko .env datoteke v korenski mapi. Spremenljivke okolja v našem primeru določajo način delovanja, bazen povezav na podatkovno bazo, koren vmesniških poizvedb ter enolični identifikator vira za podatkovno bazo. Trenutna implementacija ne dovoljuje zagona vmesnika brez pravilno nastavljenih spremenljivk okolja, čemur se lahko izognemo ob dodatni nastavitvi privzetih vrednosti spremenljivkam okolja.

```
# Get environment variables
APP_CONFIG = os.getenv("APP_CONFIG")
URL_PREFIX = os.getenv("API_URL_PREFIX")
DB_URI = str(os.getenv("API_DB_CONNECTION_STRING"))
DB_MAX_CONNECTIONS = int(os.getenv("API_DB_MAX_CONNECTIONS"))
DB_POOL_RECYCLE = int(os.getenv("API_DB_POOL_RECYCLE"))
DB_MAX_OVERFLOW = int(os.getenv("API_DB_MAX_OVERFLOW"))
DB_POOL_TIMEOUT = int(os.getenv("API_DB_POOL_TIMEOUT"))
```

*Slika 2: Spremenljivke okolja*

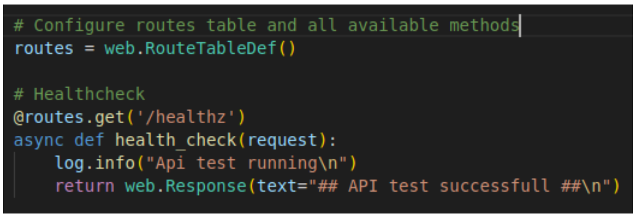
Določitev različiče izgradnje nastavi nivo beleženja podatkov med delovanjem vmesnika ter parametre bazena povezav na podatkovno bazo. Ločimo dve različici izgradnje: produkcijska različica ter razvojna različica. Različici izgradnje omogočata različne nastavitve delovanja vmesnika pred samo inicializacijo ter s tem odpravita neželene napake ali napačno beleženje delovanja vmesnika.



```
if __name__ == '__main__':
    # Set up operation mode for server
    if APP_CONFIG == 'dev':
        # Development build
        logging.basicConfig(level=logging.DEBUG)
        log = logging.getLogger()
        log.info("Running in development config")

    elif APP_CONFIG == 'prod':
        # Production build
        logging.basicConfig(level=logging.INFO)
        log = logging.getLogger()
        log.info("Running in production config")

    else:
        # If APP_CONFIG env variable is not set abort start
        logging.basicConfig(level=logging.INFO)
        log = logging.getLogger()
        log.info("Environment variable APP_CONFIG is not set
        sys.exit(1)
```

*Slika 3: Določitev različice izgradnje*

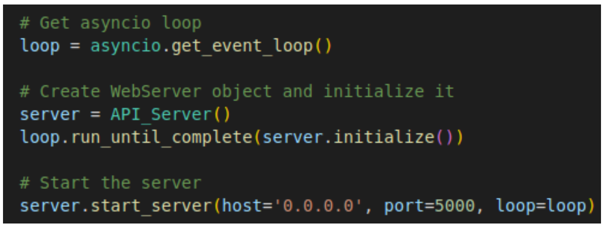
Statične poti so sprva določene z usmerjevalno tabelo in dekoratorji poti, ki jim določijo krmilnike dogodkov. Vsak krmilnik sprejme različne argumente, a v splošnem smo podatke sprejemali v JSON formatu, saj je le-ta enostaven za uporabo ter omogoča dobro preverjanje napak v podatkih poizvedb.



```
# Configure routes table and all available methods
routes = web.RouteTableDef()

# Healthcheck
@routes.get('/healthz')
async def health_check(request):
    log.info("Api test running\n")
    return web.Response(text="## API test successfull ##\n")
```

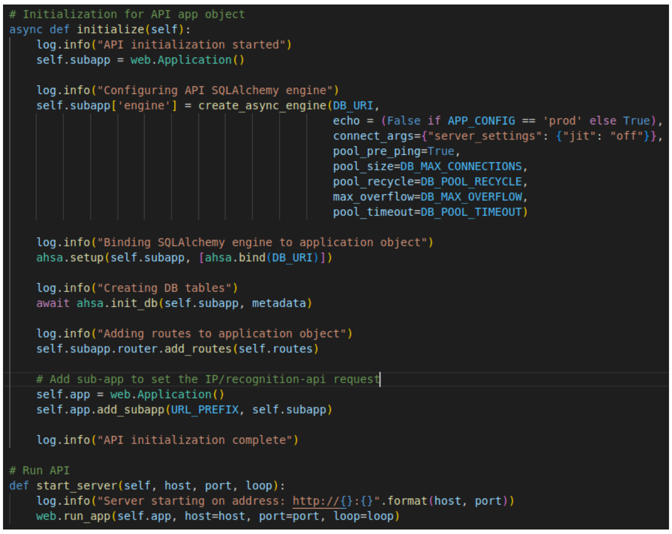
*Slika 4: Primer določitve statične poti*

Inicializacija strežniške strani se začne s pridobitvijo zanke dogodkov. Nato se izvede inicializacija, ki zajame dodajanje dodatnih atributov objektu aplikacijske storitve. V našem primeru je to inicializacija modela podatkovne baze ("db_model"), statičnih poti ter korena vmesniških poizvedb. Zagon vmesnika se zgodi po inicializaciji ter teče v neskončnost pod pogojem, da ga ne prekinemo.



```
# Get asyncio loop
loop = asyncio.get_event_loop()

# Create WebServer object and initialize it
server = API_Server()
loop.run_until_complete(server.initialize())

# Start the server
server.start_server(host='0.0.0.0', port=5000, loop=loop)
```

*Slika 5: Pridobitev zanke dogodkov ter zagon vmesnika*



```
# Initialization for API app object
async def initialize(self):
    log.info("API initialization started")
    self.subapp = web.Application()

    log.info("Configuring API SQLAlchemy engine")
    self.subapp['engine'] = create_async_engine(DB_URI,
                                                echo = (False if APP_CONFIG == 'prod' else True),
                                                connect_args={"server_settings": {"jit": "off"}},
                                                pool_pre_ping=True,
                                                pool_size=DB_MAX_CONNECTIONS,
                                                pool_recycle=DB_POOL_RECYCLE,
                                                max_overflow=DB_MAX_OVERFLOW,
                                                pool_timeout=DB_POOL_TIMEOUT)

    log.info("Binding SQLAlchemy engine to application object")
    ahsa.setup(self.subapp, [ahsa.bind(DB_URI)])

    log.info("Creating DB tables")
    await ahsa.init_db(self.subapp, metadata)

    log.info("Adding routes to application object")
    self.subapp.router.add_routes(self.routes)

    # Add sub-app to set the IP/recognition-api request
    self.app = web.Application()
    self.app.add_subapp(URL_PREFIX, self.subapp)

    log.info("API initialization complete")

# Run API
def start_server(self, host, port, loop):
    log.info("Server starting on address: http://{}:{}".format(host, port))
    web.run_app(self.app, host=host, port=port, loop=loop)
```

*Slika 6: Inicializacija vmesnika*

## Nivo podatkovnega dostopa

Nivo podatkovnega dostopa določa logiko delovanja različnih poizvedb na vmesnik. V splošnem je sestavljen iz poizvedbe na podatkovno bazo, primerne obdelave podatkov in vrnitve ali je bila izbrana metoda uspešno oziroma neuspešno zaključena.

Za razpoznavanje vzorcev sta relevantni funkciji:

- *principalComponentAnalysis*
- *optimizedPrincipalComponentAnalysis*

Obe funkciji v splošnem izvajati operacijo analize glavnih komponent s pomočjo knjižnice *scikit-learn*.

Analiza glavnih komponent je statistična metoda obdelave podatkov, ki omogoča zmanjšanje dimenzije vzorcev ob ohranjanju največje količine splošne informacije za razvrščanje le-teh. Postopek je v grobem sestavljen iz naslednjih korakov:

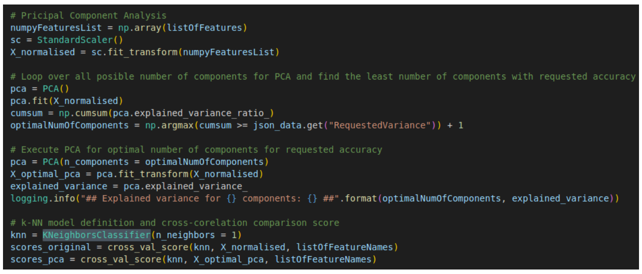
- Normalizacija vzorcev
- Izračun kovariančne matrike normaliziranih vzorcev
- Izračun lastnih vrednosti in lastnih vektorjev kovariančne matrike
- Izbira končnega števila komponent vzorcev, ki je enako ali manjše številu izbranih lastnih vrednosti/vektorjev razporejenih od največjega do najmanjšega (Najprej izberemo večje nato manjše vrednosti)
- Pretvorba normaliziranih vzorcev v nove vzorce z manjšo dimenzijo

V našem primeru normalizacijo podatkov izvede razred *StandardScaler*, ki osnovnim vzorcem odšteje povprečno vrednost ter rezultat deli s standardnim odklonom vzorcev.

Sledi uporaba *PCA* razreda, ki izvede izračun kovariančne matrike vzorcev, določi lastne vrednosti oziroma vektorje preko SVD algoritma ter izvede pretvorbo v izbrani N-dimenzijski prostor z uporabo N-največjih lastnih vrednosti. Sledi izračun variance, ki določi kako velik vpliv ima na osnovne normalizirane podtake posamezna komponenta. Izračuna se kot razmerje med vsoto izbranih lastnih vrednosti za zmanjšanje dimenzije ter vsoto vseh lastnih vrednosti.

Po zgornjem koraku se funkciji *principalComponentAnalysis* in *optimizedPrincipalComponentAnalysis* razčlenita točno zaradi variance. Ne-optimizirana funkcija izbira komponente po vrsti, medtem ko optimizirana izbere tisto kombinacijo komponent, ki predstavlja največjo varianco. Večja varianca torej omogoča bolj natančno razpoznavanje po izvedbi analize.

Sledi preizkus med osnovnimi normaliziranimi podatki ter dimenzijsko reduciranimi normaliziranimi podatki s pomočjo razpoznavalnika enega najbližjega soseda (1-NN). Za razpoznavalnik uporabimo razred *KNeighborsClassifier*, kateremu podamo argument *n_neighbors* enak ena, saj s tem določimo število najbližjih sosedov, ki jih bo razpoznavalnik upošteval.

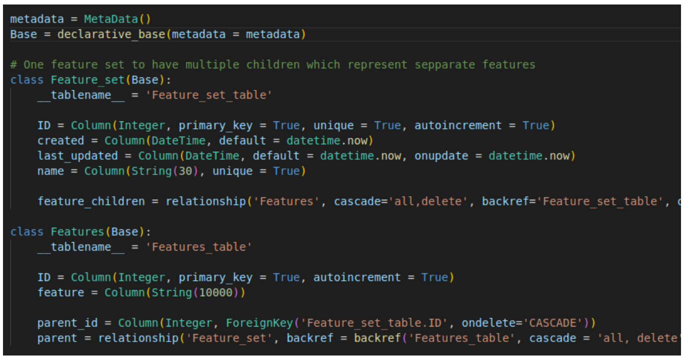


```python
# Pricipal Component Analysis
numpyFeaturesList = np.array(listOfFeatures)
sc = StandardScaler()
X_normalised = sc.fit_transform(numpyFeaturesList)

# Loop over all posible number of components for PCA and find the least number of components with requested accuracy
pca = PCA()
pca.fit(X_normalised)
cumsum = np.cumsum(pca.explained_variance_ratio_)
optimalNumOfComponents = np.argmax(cumsum >= json_data.get("RequestedVariance")) + 1

# Execute PCA for optimal number of components for requested accuracy
pca = PCA(n_components = optimalNumOfComponents)
X_optimal_pca = pca.fit_transform(X_normalised)
explained_variance = pca.explained_variance_
logging.info("## Explained variance for {} components: {} ##".format(optimalNumOfComponents, explained_variance))

# k-NN model definition and cross-corelation comparison score
knn = KNeighborsClassifier(n_neighbors = 1)
scores_original = cross_val_score(knn, X_normalised, listOfFeatureNames)
scores_pca = cross_val_score(knn, X_optimal_pca, listOfFeatureNames)
```

*Slika 6: Izsek funkcije principalComponentAnalysis*

## 3 Relacijska podatkovna baza

Strukturo relacijske podatkovne baze ("db_model") določijo sami podatki, ki jih uporabljamo. Vsaka podatkovna zbirka vsebuje več vzorcev, torej lahko sklepamo, da bo izbrana shema v obliki eden proti mnogim (*one-to-many relationship database*). V našem primeru ustvarimo znotraj podatkovne baze dve tabeli, prvo za imena različnih podatkovnih zbirk ter drugo za posamezne vzorce v zbirkah. Vsakemu elementu v drugi tabeli moramo torej ustvariti referenco na element v prvi tabeli, saj bomo le-tako lahko vedeli, kateri vzorec pripada posamezni zbirki v podatkovni bazi. Prav tako je potrebno upoštevati kaskadno brisanje vzorcev, saj bo v primeru izbrisa podatkovne zbirke iz prve tabele potrebno izbrisati vse vzorce iz druge tabele, ki se nanašajo na izbrano podatkovno zbirko.



```python
metadata = MetaData()
Base = declarative_base(metadata = metadata)

# One feature set to have multiple children which represent sepparate features
class Feature_set(Base):
    __tablename__ = 'Feature_set_table'

    ID = Column(Integer, primary_key = True, unique = True, autoincrement = True)
    created = Column(DateTime, default = datetime.now)
    last_updated = Column(DateTime, default = datetime.now, onupdate = datetime.now)
    name = Column(String(30), unique = True)

    feature_children = relationship('Features', cascade='all,delete', backref='Feature_set_table', 

class Features(Base):
    __tablename__ = 'Features_table'

    ID = Column(Integer, primary_key = True, autoincrement = True)
    feature = Column(String(10000))

    parent_id = Column(Integer, ForeignKey('Feature_set_table.ID', ondelete='CASCADE'))
    parent = relationship('Feature_set', backref = backref('Features_table', cascade = 'all, delete
```

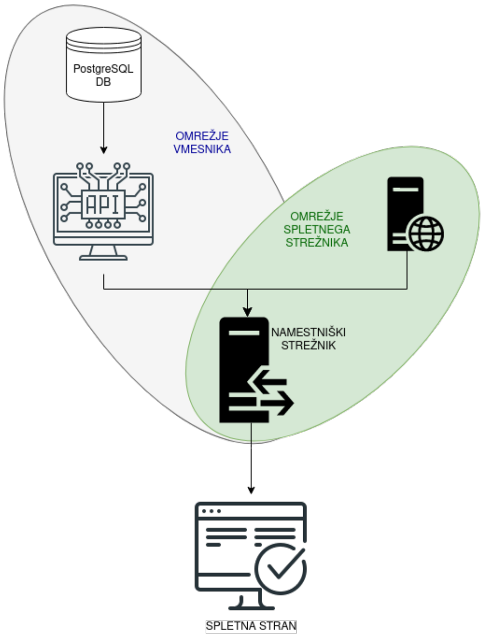
*Slika 7: Model podatkovne baze*

## 4 Postavitev storitve

Za postavitev storitve smo uporabili orodje Docker, ki omogoči optimizirano delovanje aplikacij glede na obremenjenost delujoče storitve. Prav tako smo za ažurnost knjižnic v Pythonu uporabili orodje Poetry, ki omogoči določitev specifičnih izdaj knjižnic ter tako izloči možnost kasnejše programske nekompatibilnosti. Znotraj orodja Docker smo storitev delili na sledeče podenote:

- Vmesnik za namensko programiranje
- Podatkovna baza
- Spletni strežnik (Narejena zasnova)
- Namestniški strežnik

Vsaka izmed podenot ima svoj zagonski cikel ter je odvisna od ostalih. Poleg osnovnih treh enot (vmesnik, podtakovna baza in spletni strežnik) smo dodali tudi namestniški strežnik, ki predstavlja dodaten nivo varnosti v primeru izpostavitve vmesnika internetu ter hitrejše nalaganje spletnega vmesnika s pomočjo statične hrambe določenih skript. Prav tako je bilo potrebno postaviti pravilno omrežje med aplikacijami samimi ter omejiti dostop do določenih.



*Slika 8: Omrežje postavitve storitve*

## 5 Preizkušanje delovanja vmesnika

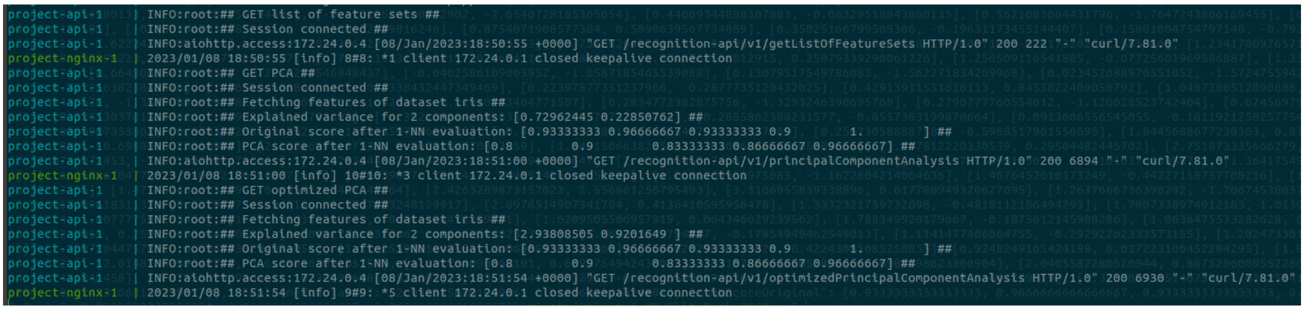
Vmesnik smo preizkusili ročno s pomočjo terminala z uporabo modula *curl*. Vse preizkusne poizvedbe so zbrane znotraj datoteke *test_requests.txt*. Z njihovimi ekvivalenti je mogoče vmesnik uporabiti na različnih zbirkah ter preveriti njegovo delovanje.

```
timotej@timotej-HP-EliteBook-8560w:~$ curl --request GET http://127.0.0.1:5000/recognition-api/v1/getListOfFeatureSets
{"status": 200, "message": {"FeatureSets": ["iris", "dryBean"]}}timotej@timotej-HP-EliteBook-8560w:~$
timotej@timotej-HP-EliteBook-8560w:~$
timotej@timotej-HP-EliteBook-8560w:~$ curl --request GET --header "Content-Type: application/json" --data '{"FeatureSet": "iris", "NumOfComponents": 2}' http://127.0.0.1:5000/recognition-api/v1/principal
ComponentAnalysis
{"status": 200, "message": {"FeatureSet": "iris", "X_pca": [[-2.264702808807597, 0.4800265965209889], [-2.0809611519657687, -0.6741335566053532], [-2.3642290538903006, -0.3419080238846763], [-2.299384217
042709, -0.597394507674676], [-2.389842166313845, 0.6468353829020271], [-2.0756309481765123, 1.4891775233211668], [-2.4440288351341533, 0.0476441976300141], [-2.2328471588720156, 0.22314807268959172], [-
2.334640477907622, -1.1153276754616666], [-2.1843281749339427, -0.469013561402376], [-2.1663101007013226, 1.0436906530538599], [-2.326130866442699, 0.13307833523923146], [-2.2184508988224096, -0.72867616
53165712], [-2.633100695765229, -0.9615067291701632], [-2.1987406032666903, 1.8600571132939314], [-2.2622145316010207, 2.6862844851105923], [-2.207587695824592, 1.4836093631555727], [-2.190349509192299,
0.4888383164863276], [-1.898571995802842, 1.4050187944665495], [-2.3433690530749933, 1.1278493819084767], [-1.9143229960825678, 0.4088557077559061], [-2.2070128431947986, 0.9241214267468988], [-2.7743447
02927333, 0.4583436677529153], [-1.8186695286958492, 0.0855585262873662], [-2.227163305706639, 0.13725445536342704], [-1.9518463309003764, -0.6256185877766768], [-2.051151372729416, 0.24216355266166711],
 [-2.168577174654217, 0.5271495253082675], [-2.139563451301332, 0.31321781013995187], [-2.2652614931542416, -0.33773190376048084], [-2.1401221356479847, -0.5045406901415186], [-1.8315947706760285, 0.4236
9506760378595], [-2.6149479358589356, 1.793575856104429], [-2.446177391696515, 2.150727877392924], [-2.109974875318653, -0.46020184143703763], [-2.207808899078266, -0.20610739768843697], [-2.045146206754
2012, 0.6615581114631083], [-2.5273319131704866, 0.5922927741908095], [-2.429632575084547, -0.9041800403761483], [-2.169710711630664, 0.26887896143547085], [-2.2864751433456707, 0.4417153876990502], [-1.
8581224563735714, -2.337415157553349], [-2.5536383956143567, -0.4791006901223142], [-1.9644476837637412, 0.4723266677192611], [-2.1370590058116234, 1.1422292620394092], [-2.06974429959183, -0.71105272538
58943], [-2.3847331657782633, 1.1204297019845368], [-2.3943763142196337, -0.38624687258915746], [-2.2294465479426746, 0.9979597643079806], [-2.2038334355191305, 0.009216357521276225], [1.101781183052949,
 0.8629724182621592], [0.7313374253960883, 0.5946147256694249], [1.2409793195158327, 0.6162976544374987], [0.40748305881738545, -1.754403989323411], [1.0754747006090792, -0.20842104605096537], [0.3886873
3653566506, -0.5932836359900754], [0.7465297413291615, 0.7730193120985968], [-0.4873227421256397, -1.8524290868575748], [0.9279016383549463, 0.03222607789115391], [0.011426188736980133, -1.03401827512944
04], [-0.1101962800062902, -2.6540728185365654], [0.44069344898307883, -0.06329518843800135], [0.5621083064431796, -1.7647243806169455], [0.7195618886754972, -0.18622460583150585], [-0.03335470317877195
5, -0.4390032099816246], [0.8754071908577384, 0.5090639567734089], [0.35025166799508306, -0.19631173455144407], [0.15881004754797148, -0.7920957424327216], [1.2250936335624325, -1.6222438030915018], [0.1
649178993863276, -1.302609230295773], [0.7376826487712588, 0.3965715619602392], [0.4762871909409719, -0.41732028121355086], [1.23417809765715, -0.933257287992788], [0.6328581997098223, -0.4163877208890
997], [0.7026611831361831, -0.06341181972480005], [0.8742736538812915, 0.25079339290061226], [1.256509116541885, -0.07725601969586887], [1.358405121440633, 0.33131168179089837], [0.6648003672253953, -0.22
5927854694843 7], [-0.0402586109005952, -1.0587185465539088], [0.13079517549786085, -1.562271834209968], [0.023452688970551052, -1.5724755942167052], [0.24153827295451125, -0.7772563825848416], [1.0610946
088426148, -0.6338432447349469], [0.22397877351237966, -0.2877735120432025], [0.42913911551616113, 0.8455822409050792], [1.0487280512090886, 0.5220517968629429], [1.04453138439628, -1.382988719190782], [
0.06958832111642316, -0.2195033464771507], [0.2834772382875756, -1.3293246390695768], [0.2790777760554612, -1.120028523742404], [0.6245697914985721, 0.02492302925401263], [0.33653037013143616, -0.988404
0176703606], [-0.3621833846193833, -2.0192378732386125], [0.2885862388231577, -0.8557303199870664], [0.0913606556545055, -0.18119212582577543], [0.2277168655347003, -0.38492008098735386], [0.576388288653
4792, -0.15487359721655883], [-0.4476670190286118, -1.5437920343977565], [0.256730588887585, -0.5988517961556695], [1.8445688677230303, 0.8704213123248237], [1.1578816132057803, -0.6988698623306904], [2.
20526679107538, 0.5620104770083559], [1.4401506638275403, -0.046987588105806555], [1.867812220330539, 0.29504482445702], [2.751873335666279, 0.8004092010275423], [0.36701768786072453, -1.5615028914765061
], [2.3024394446251994, 0.42006557964277647], [2.006686467676607, -0.711438653547159], [2.2597773490125026, 1.9210103764598871], [1.364175492186009, 0.6927564544903874], [1.6026786704779319, -0.421700449
772617], [1.8839007017032443, 0.4192496506051238], [1.2601150991975083, -1.1622604214064636], [1.4676452010173249, -0.44227158737708216], [1.5900773176145668, 0.6762448057233206], [1.4714314611333192, 0.
25562182447147064], [2.4263289873157023, 2.556661250795493], [3.3106955839338896, 0.017780949320627095], [1.2637666736398292, -1.706745380376268], [2.037716301469407, 0.9104674096183113], [0.977980734249
4219, -0.5717643248129917], [2.8976514907341704, 0.4136410595956478], [1.3332321759732098, -0.4818112186494293], [1.7007338974912183, 1.0139218673227919], [1.9543267058530724, 1.007777596153453], [1.1751
036315549332, -0.316394472309791], [1.0209505506957919, 0.064346029239562], [1.788349920179667, -0.1873612145908286], [1.8636475533282628, 0.5622907258861448], [2.4359537279227053, 0.25928443314428046],
[2.304927721831766, 2.6263234682323793], [1.862703219794957, -0.17854949462549013], [1.1141477406864755, -0.29292262333573155], [1.2024733016783917, -0.8113152708396693], [2.79877044757811, 0.85680332949
71058], [1.5762559101947553, 1.068581107320808], [1.3462921036270628, 0.42243061085250855], [0.9248249165424198, 0.017223100452284295], [1.8520450517676719, 0.6761281744365221], [2.0148104299548772, 0.61
38856369235757], [1.901784090262191, 0.6895754942430032], [1.1578816132057803, -0.6988698623306904], [2.0405582280520944, 0.8675206009552286], [1.9981470959523775, 1.0491687471841458], [1.870503292956411
8, 0.38696608166572605], [1.5645804830303291, -0.8966868088965264], [1.521170499627839, 0.2690691442779519], [1.3727877895140739, 1.0112544185267935], [0.9606560300371284, -0.024331668169399277]], "Expla
inedVariance": [0.7296244541329987, 0.2285076178670178], "1-NNScoreOriginal": [0.9333333333333333, 0.9666666666666667, 0.9333333333333333, 0.9, 1.0], "1-NNScorePCA": [0.8, 0.9, 0.8333333333333334, 0.8666
666666666667, 0.9666666666666667]}}timotej@timotej-HP-EliteBook-8560w:~$
```

*Slika 9: Enostavna poizvedba na vmesnik ter poizvedba principalComponentAnalysis na strežnik ter odziva v JSON formatu*

```
timotej@timotej-HP-EliteBook-8560w:~$ curl --request GET --header "Content-Type: application/json" --data '{"FeatureSet": "iris", "RequestedVariance": 0.95}' http://127.0.0.1:5000/recognition-api/v1/opti
mizedPrincipalComponentAnalysisoptimizedPrincipalComponentAnalysis
{"status": 200, "message": {"FeatureSet": "iris", "OptimalNumOfComponents": 2, "X_optimal_pca": [[-2.264702808807597, 0.4800265965209889], [-2.0809611519657687, -0.6741335566053532], [-2.3642290538903006
, -0.3419080238846763], [-2.299384217042709, -0.597394507674676], [-2.389842166313845, 0.6468353829020271], [-2.0756309481765123, 1.4891775233211668], [-2.4440288351341533, 0.0476441976300141], [-2.23284
71588720156, 0.22314807268959172], [-2.334640477907622, -1.1153276754616666], [-2.1843281749339427, -0.469013561402376], [-2.1663101007013226, 1.0436906530538599], [-2.326130866442699, 0.1330783352392314
6], [-2.2184508988224096, -0.7286761653165712], [-2.633100695765229, -0.9615067291701632], [-2.1987406032666903, 1.8600571132939314], [-2.2622145316010207, 2.6862844851105923], [-2.207587695824592, 1.483
6093631555727], [-2.190349509192299, 0.4888383164863276], [-1.898571995802842, 1.4050187944665495], [-2.3433690530749933, 1.1278493819084767], [-1.9143229960825678, 0.4088557077559061], [-2.2070128431947
986, 0.9241214267468988], [-2.774344702927333, 0.4583436677529153], [-1.8186695286958492, 0.0855585262873662], [-2.227163305706639, 0.13725445536342704], [-1.9518463309003764, -0.6256185877766768], [-2.0
51151372729416, 0.24216355266166711], [-2.168577174654217, 0.5271495253082675], [-2.139563451301332, 0.31321781013995187], [-2.2652614931542416, -0.33773190376048084], [-2.1401221356479847, -0.5045406901
415186], [-1.8315947706760285, 0.42369506760378595], [-2.6149479358589356, 1.793575856104429], [-2.446177391696515, 2.150727877392924], [-2.109974875318653, -0.46020184143703763], [-2.207808899078266, -0
.20610739768843697], [-2.0451462067542012, 0.6615581114631083], [-2.5273319131704866, 0.5922927741908095], [-2.429632575084547, -0.9041800403761483], [-2.169710711630664, 0.26887896143547085], [-2.286475
1433456707, 0.4417153876990502], [-1.8581224563735714, -2.337415157553349], [-2.5536383956143567, -0.4791006901223142], [-1.9644476837637412, 0.4723266677192611], [-2.1370590058116234, 1.1422292620394092
], [-2.06974429959183, -0.7110527253858943], [-2.3847331657782633, 1.1204297019845368], [-2.3943763142196337, -0.38624687258915746], [-2.2294465479426746, 0.9979597643079806], [-2.2038334355191305, 0.009
216357521276225], [1.101781183052949, 0.8629724182621592], [0.7313374253960883, 0.5946147256694249], [1.2409793195158327, 0.6162976544374987], [0.40748305881738545, -1.754403989323411], [1.07547470060907
92, -0.20842104605096537], [0.38868733653566506, -0.5932836359900754], [0.7465297413291615, 0.7730193120985968], [-0.4873227421256397, -1.8524290868575748], [0.9279016383549463, 0.03222607789115391], [0.
011426188736980133, -1.0340182751294404], [-0.1101962800062902, -2.6540728185365654], [0.44069344898307883, -0.06329518843800135], [0.5621083064431796, -1.7647243806169455], [0.7195618886754972, -0.1862
2460583150585], [-0.033354703178771955, -0.4390032099816246], [0.8754071908577384, 0.5090639567734089], [0.35025166799508306, -0.19631173455144407], [0.15881004754797148, -0.7920957424327216], [1.2250936
335624325, -1.6222438030915018], [0.1649178993863276, -1.302609230295773], [0.7376826487712588, 0.3965715619602392], [0.4762871909409719, -0.41732028121355086], [1.23417809765715, -0.933257287992788], [
0.6328581997098223, -0.4163877208890997], [0.7026611831361831, -0.06341181972480005], [0.8742736538812915, 0.25079339290061226], [1.256509116541885, -0.07725601969586887], [1.358405121440633, 0.331311681
79089837], [0.6648003672253953, -0.2259278546948437], [-0.0402586109005952, -1.0587185465539088], [0.13079517549786085, -1.562271834209968], [0.023452688970551052, -1.5724755942167052], [0.24153827295451
125, -0.7772563825848416], [1.0610946088426148, -0.6338432447349469], [0.22397877351237966, -0.2877735120432025], [0.42913911551616113, 0.8455822409050792], [1.0487280512090886, 0.5220517968629429], [1.0
4453138439628, -1.382988719190782], [0.06958832111642316, -0.2195033464771507], [0.2834772382875756, -1.3293246390695768], [0.2790777760554612, -1.120028523742404], [0.6245697914985721, 0.02492302925401
263], [0.33653037013143616, -0.9884040176703606], [-0.3621833846193833, -2.0192378732386125], [0.2885862388231577, -0.8557303199870664], [0.0913606556545055, -0.18119212582577543], [0.2277168655347003, -
0.38492008098735386], [0.5763882886534792, -0.15487359721655883], [-0.4476670190286118, -1.5437920343977565], [0.256730588887585, -0.5988517961556695], [1.8445688677230303, 0.8704213123248237], [1.157881
6132057803, -0.6988698623306904], [2.20526679107538, 0.5620104770083559], [1.4401506638275403, -0.046987588105806555], [1.867812220330539, 0.29504482445702], [2.751873335666279, 0.8004092010275423], [0.3
6701768786072453, -1.5615028914765061], [2.3024394446251994, 0.42006557964277647], [2.006686467676607, -0.711438653547159], [2.2597773490125026, 1.9210103764598871], [1.364175492186009, 0.692756454490387
4], [1.6026786704779319, -0.421700449772617], [1.8839007017032443, 0.4192496506051238], [1.2601150991975083, -1.1622604214064636], [1.4676452010173249, -0.44227158737708216], [1.5900773176145668, 0.67624
48057233206], [1.4714314611333192, 0.25562182447147064], [2.4263289873157023, 2.556661250795493], [3.3106955839338896, 0.017780949320627095], [1.2637666736398292, -1.706745380376268], [2.037716301469407,
 0.9104674096183113], [0.9779807342494219, -0.5717643248129917], [2.8976514907341704, 0.4136410595956478], [1.3332321759732098, -0.4818112186494293], [1.7007338974912183, 1.0139218673227919], [1.95432670
58530724, 1.007777596153453], [1.1751036315549332, -0.316394472309791], [1.0209505506957919, 0.064346029239562], [1.788349920179667, -0.1873612145908286], [1.8636475533282628, 0.5622907258861448], [2.435
9537279227053, 0.25928443314428046], [2.304927721831766, 2.6263234682323793], [1.862703219794957, -0.17854949462549013], [1.1141477406864755, -0.29292262333573155], [1.2024733016783917, -0.81131527083966
93], [2.79877044757811, 0.8568033294971058], [1.5762559101947553, 1.068581107320808], [1.3462921036270628, 0.42243061085250855], [0.9248249165424198, 0.017223100452284295], [1.8520450517676719, 0.6761281
744365221], [2.0148104299548772, 0.6138856369235757], [1.901784090262191, 0.6895754942430032], [1.1578816132057803, -0.6988698623306904], [2.0405582280520944, 0.8675206009552286], [1.9981470959523775, 1.
0491687471841458], [1.8705032929564118, 0.38696608166572605], [1.5645804830303291, -0.8966868088965264], [1.521170499627839, 0.2690691442779519], [1.3727877895140739, 1.0112544185267935], [0.960656030037
1284, -0.024331668169399277]], "ExplainedVariance": [2.938085050199996, 0.9201649041624882], "1-NNScoreOriginal": [0.9333333333333333, 0.9666666666666667, 0.9333333333333333, 0.9, 1.0], "1-NNScorePCA": [
0.8, 0.9, 0.8333333333333334, 0.8666666666666667, 0.9666666666666667]}}timotej@timotej-HP-EliteBook-8560w:~$
```

*Slika 10: Poiveddba optimizedPrincipalComponentAnalysis na strežnik ter odziv v JSON formatu*



```
project-api-1    | INFO:root:## GET list of feature sets ##
project-api-1    | INFO:root:## Session connected ##
project-api-1    | INFO:aiohttp.access:172.24.0.4 [08/Jan/2023:18:50:55 +0000] "GET /recognition-api/v1/getListOfFeatureSets HTTP/1.0" 200 222 "-" "curl/7.81.0"
project-nginx-1  | 2023/01/08 18:50:55 [info] 8#8: *1 client 172.24.0.1 closed keepalive connection
project-api-1    | INFO:root:## GET PCA ##
project-api-1    | INFO:root:## Session connected ##
project-api-1    | INFO:root:## Fetching features of dataset iris ##
project-api-1    | INFO:root:## Explained variance for 2 components: [0.72962445 0.22850762] ##
project-api-1    | INFO:root:## Original score after 1-NN evaluation: [0.93333333 0.96666667 0.93333333 0.9        1.        ] ##
project-api-1    | INFO:root:## PCA score after 1-NN evaluation: [0.8        0.9        0.83333333 0.86666667 0.96666667] ##
project-api-1    | INFO:aiohttp.access:172.24.0.4 [08/Jan/2023:18:51:00 +0000] "GET /recognition-api/v1/principalComponentAnalysis HTTP/1.0" 200 6894 "-" "curl/7.81.0"
project-nginx-1  | 2023/01/08 18:51:00 [info] 10#10: *3 client 172.24.0.1 closed keepalive connection
project-api-1    | INFO:root:## GET optimized PCA ##
project-api-1    | INFO:root:## Session connected ##
project-api-1    | INFO:root:## Fetching features of dataset iris ##
project-api-1    | INFO:root:## Explained variance for 2 components: [2.93808505 0.9201649 ] ##
project-api-1    | INFO:root:## Original score after 1-NN evaluation: [0.93333333 0.96666667 0.93333333 0.9        1.        ] ##
project-api-1    | INFO:root:## PCA score after 1-NN evaluation: [0.8        0.9        0.83333333 0.86666667 0.96666667] ##
project-api-1    | INFO:aiohttp.access:172.24.0.4 [08/Jan/2023:18:51:54 +0000] "GET /recognition-api/v1/optimizedPrincipalComponentAnalysis HTTP/1.0" 200 6930 "-" "curl/7.81.0"
project-nginx-1  | 2023/01/08 18:51:54 [info] 9#9: *5 client 172.24.0.1 closed keepalive connection
```

*Slika 11: Primer delovanja strežnika ob zgornjih treh poizvedbah*

## 6 Povzetek

Trenutno je izdelan prototip delovanja storitve (Vmesnika ter spletnega strežnika), ki ga je kasneje mišljeno izboljšati in razširiti na poljubno velikost (Zavzame 8 izbirnih projektov). Glede same analize glavnih komponent pa se je izkazalo, da je z reduciranjem komponent mogoče doseči le omejeno količino razpoznanih podatkov. Najboljši rezultat smo dobili pri 2 komponentah, kjer je najnižja razpoznana vrednost po reduciranju bila 80% z uporabo osnovnih podatkov pa smo dosegli najnižjo vrednost 90%.

## 7 Viri in literatura

- S. Dobrišek: Razpoznavanje vzorcev, gradivo za predavanja, FE 2022
- K. Grm: Razpoznavanje vzorcev, gradivo za laboratorijske vaje, FE 2022
- Dokumentacija orodja Docker
- Dokumentacija namestniškega strežnika NGINX
- Dokumentacija o jezikih HTML5, CSS ter JS
- Dokumentacija uporabljenih knjižnic v Pythonu:
  - Vmesnik - "recognition_api": *logging*, *os*, *sys*, *asyncio*, *asyncpg*, *aiohttp*, *aiohttp_sqlalchemy*, *sqlalchemy*
  - Vmesnik - "recognition_DAL": *logging*, *numpy*, *sqlalchemy*, *sklearn*
  - Vmesnik - "db_model": *datetime*, *sqlalchemy*
  - Spletni strežnik - "webserver": *logging*, *os*, *sys*, *asyncio*, *aiohttp*