

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU  
RAČUNARSTVO

IVONA REBROVIĆ

PYTHON WEB APLIKACIJA

ZAVRŠNI RAD

ČAKOVEC, 2015.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU  
RAČUNARSTVO

IVONA REBROVIĆ

PYTHON WEB APLIKACIJA  
PYTHON WEB APPLICATION

ZAVRŠNI RAD

Mentor:

mr. sc. Bruno Trstenjak

ČAKOVEC, 2015.

## ZAHVALA

Zahvalila bih se svim profesorima Međimurskog veleučilišta u Čakovcu i mentoru mr.sc. Bruni Trstenjaku na svom pruženom znanju i podršci tokom studiranja te pomoći kod izrade i realizacije ovog završnog rada.

## SAŽETAK

*Cilj ovog završnog rada je prikazati karakteristike i mogućnosti Pythona te rad Python web aplikacije. Python web aplikacija rađena je u Eclipse Java EE IDE razvojnom okruženju. Korištena je verzija Luna Release 4.4.0. Eclipse nudi šest vrsta alata za Python razvojnu okolinu. U razvoju aplikacije korišten je PyDev alat te Django open-source web razvojna okolina. Django je pisan u Pythonu te koristi model - pogled - predložak (MVT) arhitekturu. U svom radu, Python web aplikacija upotrebom SciKit razvojnog paketa i Naive Bayes-ovog algoritma izračunava vjerojatnost uspješnosti studenta u studiranju. SciKit razvojni paket sadrži biblioteku raznih klasa iz područja strojnog učenja i Data mininga. Klase omogućuju implementaciju raznih metoda i algoritama za klasifikaciju podataka te određivanje regresije među podacima i atributima. Strojno učenje je programiranje računala na način da optimizira kriterije uspješnosti na temelju podatkovnih primjera ili prethodnog iskustva. Naive Bayes radi na principu izračunavanja vjerojatnosti hipoteze koja proizlazi iz pojedinih dijelova dokaza, kada su dane samo vjerojatnosti prema kojima dokazi proizlaze iz stvarnih posljedica (hipoteza).*

*Python web aplikacija omogućuje klasifikaciju ulaznih podataka na osnovu baze podataka, te upravljanje s podacima te iste baze podataka. U Python web aplikaciji korištena je MySQL baza podataka. Za povezivanje aplikacije s bazom podataka, web aplikacija koristi MySQL Connector Python. Aplikacija omogućuje upravljanje podacima putem Django ORM. Django ORM na temelju objektnog modela definiranog kroz models.py generira strukturu u relacijskoj bazi podataka. Baza podataka povezana je preko models.py dok je predložak (html) povezan preko views.py. Model dohvaća tražene vrijednosti iz baze podataka te ih prosljeđuje elementu views.py u aplikaciji. Views.py određuje koji će podaci biti prikazani odnosno koji će podaci biti dohvaćeni iz baze podataka i prikazani u web pregledniku.*

*U završnom radu provedena su razna testiranja u području klasifikacije podataka i kvalitete pristupa bazi podataka. Nad ostvarenim rezultatima provedena je detaljna analiza.*

**Ključne riječi:** Python, strojno učenje, klasifikacija, web aplikacija, Naive Bayes.

## SADRŽAJ

1. UVOD .....	5
2. CILJ ISTRAŽIVANJA .....	6
3. RAZVOJNA OKOLINA.....	7
4. PYTHON.....	8
4.1. Povezivanje Python aplikacije sa bazom podataka .....	12
4.2. Django i PyDev .....	13
4.3. SciKit razvojni paket .....	14
5. STROJNO UČENJE .....	15
5.1. Naive Bayes .....	17
6. BAZA PODATAKA I KLASIFIKACIJSKI PODACI.....	18
6.1. Django ORM.....	18
6.2. Atributi objekata / models.py.....	20
7. PYTHON APLIKACIJA.....	20
7.1. Moduli aplikacije .....	23
8. TESTIRANJE APLIKACIJE.....	25
9. ZAKLJUČAK .....	30
10. LITERATURA .....	31

## 1. UVOD

Programski jezik Python nastao je 1990. godine te se njegovim tvorcem smatra Guido van Rossum[1]. Python pripada skupini objektno orijentiranih i skriptnih jezika. Python je slobodan jezik koji je zbog svoje popularnosti i praktičnosti zastupljen na velikom broju platformi. Koristi se u izradi web i mobilnih aplikacija. Zbog svoje jednostavnosti, Python koriste svjetski poznate ustanove poput Google-a, Yahoo-a, NASA-e i dr. U svom radu koristi simultano prevođenje, odnosno program se prevodi u strojni jezik. To je jezik opće namjene, interpretiran te ga karakterizira izrazito pregledna sintaksa i velik broj biblioteka.

U razvoju Python web aplikacije korišten je SciKit razvojni paket kao podrška za razne metode i algoritme strojnog učenja. Aplikacija omogućuje klasifikaciju podatka upotrebom Naive Bayes algoritma. Implementacijom spomenutog algoritma, aplikacija provodi predviđanje uspješnosti u studiranju. Osim predviđanja i klasifikacije podataka, aplikacija omogućuje upravljanje podacima o studentima koji su zapisani u MySQL relacijskoj bazi podataka. Podaci korisnika dohvaćaju se iz baze podataka putem modela, dok se pregled podataka ostvaruje upotrebom pogleda ( `views.py` ). Za izvršavanje Query upita prema bazi podataka korišten je Django ORM.

## 2. CILJ ISTRAŽIVANJA

Cilj ovog završnog rada je izrada Python web aplikacije koja klasifikacijom podataka te predviđanjem određuje uspješnost odnosno završetak studija pojedinog studenta. Podaci studenata nalaze se u MySQL relacijskoj bazi. Aplikacija je razvijena u Eclipse razvojnoj okolini.

Vjerojatnost hoće li student završiti određeni studij izračunava se upotrebom Naive Bayes-ovog algoritma i SciKit razvojnog paketa. Spomenuti algoritam na osnovu ulaznih podataka o studentu, provodi proces klasifikacije i procjenjuje da li će student s tim podacima uspješno završiti studij. Podaci o studentu upisuju se u zadanoj formi i traženim atributima. Na temelju upisanih vrijednosti atributa, algoritam izračunava vjerojatnost, a zatim se ti podaci spremaju u bazu podataka.

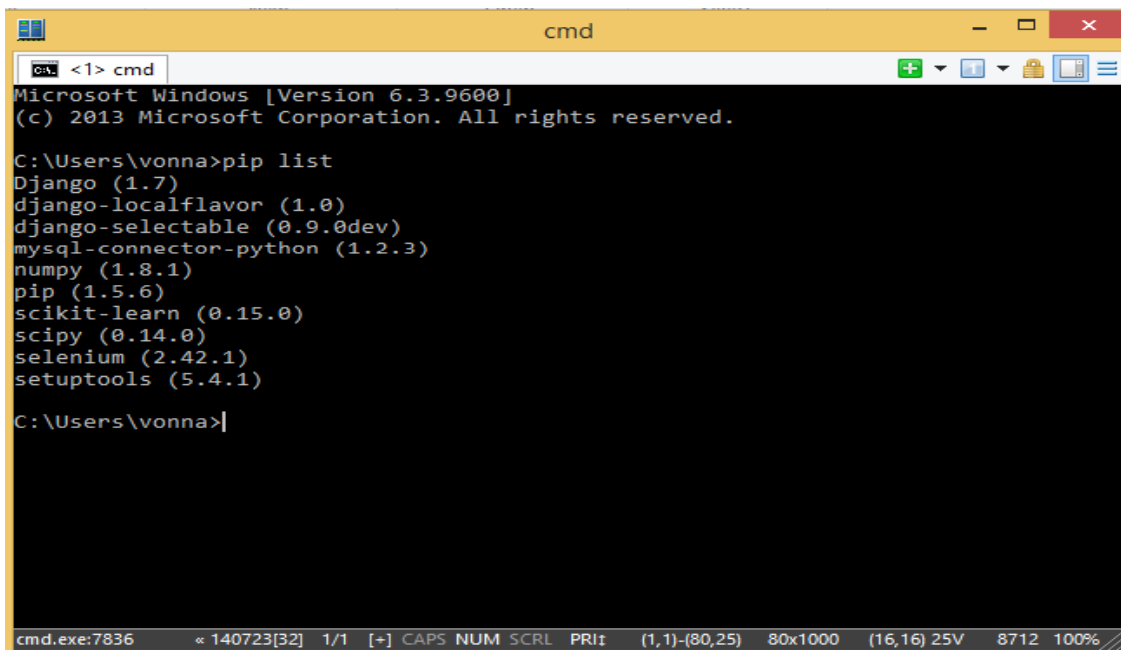
Korištenjem Django ORM-a manipulira se podacima u bazi podataka. Pomoću modela, podaci se iz baze podataka dohvaćaju te se prosljeđuju pogledima (engl. *views*).

Cilj je prikaz Python jezika i njegove primjene u području strojnog učenja te integracija programskih jezika i tehnologija u web okruženju.

### 3. RAZVOJNA OKOLINA

Integrirana razvojna okolina ( eng. *Integrated development environment - IDE* ) je programsko okruženje kojem je cilj olakšati razvoj novih programskih aplikacija. Bez razvojnih okolina teško je zamisliti razvoj složenih aplikacija.

**Slika 1.** Konzolni prikaz alata korištenih u izradi aplikacije



```
cmd
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\vonna>pip list
Django (1.7)
django-localflavor (1.0)
django-selectable (0.9.0dev)
mysql-connector-python (1.2.3)
numpy (1.8.1)
pip (1.5.6)
scikit-learn (0.15.0)
scipy (0.14.0)
selenium (2.42.1)
setuptools (5.4.1)

C:\Users\vonna>
```

Na slici 1. prikazan je sastav datoteka korištenih u izradi aplikacije. Razvoj aplikacije proveden je pomoću programa Eclipse Java EE IDE verzija Luna Release 4.4.0.

Eclipse je programska razvojna okolina pisana u Javi. Sastoji se od radne okoline i dodatnih alata koji se mogu uključiti prema potrebama korisnika. U izradi aplikacije korišten je PyDev alat. Eclipse se koristi kao urednik Python programskog koda. Postoji šest vrsta alata za Python razvojnu okolinu: Dynamic Languages Toolkit<sup>1</sup>, PyDev<sup>2</sup>, JyDT<sup>3</sup>, PythonMonkey<sup>4</sup>, EclipseColorer<sup>5</sup> i TruStudio<sup>6</sup>.

<sup>1</sup> Dynamic Languages Toolkit - Alat za proizvođače, pretraživače i krajnje korisnike koji se oslanjaju na dinamičke jezike

<sup>2</sup> PyDev - Koristi značajke kao što su automatsko dopunjavanje koda ili analiza koda. Trećerazredna ekstenzija Eclipse-a

<sup>3</sup> JyDT - Jython Development Tools je Java implementacija Pythona.



## 4. PYTHON

Programski jezik Python nastao je 1990. godine te se njegovim osnivačem smatra Guido van Rossum. Svoje zanimljivo ime dobio je po televizijskoj seriji *Monty Python's Flying Circus*. Zbog svoje fleksibilnosti i jednostavnosti kod korištenja te mogućnosti da programer koristi nekoliko stilova programiranja i zastupljenost na velikom broju platformi, stavlja ga se na popis popularnijih programskih jezika. Pisan je u C jeziku te se može proširivati dodavanjem raznih C biblioteka. Python ima veliku sličnost s programskim jezicima Pearl <sup>7</sup>, Ruby <sup>8</sup>, Smalltalk <sup>9</sup> i dr.

Python ne koristi vitičaste zagrade za definiranje bloka naredbi kao drugi programski jezici, već se uvlačenjem koda razlikuju programski blokovi. Strukturni, aspektni i objektno orijentirani stilovi programiranja u Pythonu su dopušteni jer je interpretiran jezik opće namjere i visoke razine. Bez obzira na svoju jednostavnost i dalje je relativno snažan jezik. Uz korištenje skriptnih jezika nudi napredne programske alate koji su tipični za razvoj programa namijenjeni operativnim sustavima. Na slici 2. prikazan je Python logo.

**Slika 2.** Logo Python programskog jezika



( <https://www.python.org/community/logos/> )

Zbog svoje primjene Python je već 1998. godine imao bazu od 300 000 korisnika, te ga koriste ustanove poput MIT-a, NASA-e, IBM-a, Google-a, Yahoo-a i dr. Python nije

---

<sup>4</sup> PythonMonkey - radi promjene kod modula ili klasa dok je program pokrenut

<sup>5</sup> EclipseColorer - Bazirano na Colorer-take5 biblioteci; svaka platforma ima svoj izvorni dio biblioteke

<sup>6</sup> TruStudio - napredni PHP i Python IDE. Služi za naprednu analizu i navigaciju koda

<sup>7</sup> Pearl - programski jezik opće namjene. Autor je Larry Wall

<sup>8</sup> Ruby - dinamičko, objektno orijentiran programski jezik. Nastao u Japnu 90-tih godina

<sup>9</sup> Smalltalk - dinamičko, objektno orijentirani programski jezik. Dizajniran i napravljen u obrazovnu svrhu.

donio nove revolucionarne ideje i značajke u programiranju već je ujedinio sve najbolje ideje iz drugih programskih jezika. [2]

#### ZNAČAJKE:

- Python kod sprema se u tekstualne datoteke s nastavkom `.py`.
- Za izvršavanje programa napisanom u Python jeziku, potrebno ga je prevesti. Prevođenjem programa stvara se datoteka koja sadrži međukodove (engl. *bytecode*). Ekstenzija generirane datoteke je `.pyc`. Prije pokretanja programa, međukod se prevodi na razinu računala na kojem se izvodi program.
- Uz standardne tipove podataka ima ugrađene i tipove podataka visoke razine, poput lista, n-torka<sup>10</sup> i rječnika.
- Programi pisani u Pythonu mogu se izvoditi u različitim okruženjima.
- Sintaksa jezika je jednostavna, program je vrlo pregledan i jednostavan za čitanje.

U Windows OS-u Python nudi više grafičkih rješenja za interaktivan rad. U njegovom instalacijskom paketu dolazi IDLE (engl. *Integrated DeveLopment Environment*) kojeg je razvio Rossum te Tkinter<sup>11</sup> potpora. Za jednostavniju upotrebu te grafička rješenja u interaktivnom i skriptnom radu koristi se PythonWin ( eng. *Python Editor for Windows* ) i PyScripter ( eng. *open source Python IDE* ). Prilikom izvođenja Python programa, korisnik ima mogućnost korištenja nekoliko načina rada:

- **Skriptni rad** - Kod spremanja programa u skripte i pozivanja Unix skripte iz sistemskih linija koristi se standardni tekst editor ili Python orijentirano okruženje
- **Programski moduli** - Pozivom modula i Python interpretera, moduli s Python naredbama izvode se iz systemske jezgre. Primjer modula:

```
% python moj_modul.py
```

- **Umetnuti** ( eng. *embedded* ) **kod** - Korištenjem *Python runtime API*<sup>12</sup>, Python kod može se izvoditi u obliku izvornog teksta naredbi unutar drugog programa.

---

<sup>10</sup> n - torka - Nepromjenjiv niz članova. Objekti istih ili različitih parova.

<sup>11</sup> Tkinter - standardni Python-ov paket grafičkog korisničkog sučelja

<sup>12</sup> API - Aplikacijsko programsko sučelje

**Osnovne naredbe u Python-u:**

NAREDBA	OBJAŠNJENJE
PRINT	Naredba za ispis. Praktična i jednostavna naredba, ispisuje vrijednosti u tekstualnom obliku. Implicitno izbacuje razmak između izraza i također implicitno uključuje novi red nakon posljednjeg izraza, osim u situaciji kada iza posljednjeg izraza slijedi zarez.
RETURN	Naredba se koristi jedino u funkciji te nakon nje slijedi neki izraz. Nakon izvršavanje naredbe, funkcija završava te se dobiva povratna vrijednost izraza. Dolaskom do kraja svoje funkcije ili izvršenjem naredbe bez pridruženog izraza, funkcija vraća vrijednost <i>None</i> . Naredba se ne piše na kraju koda.
FOR	Naredba ostvaruje iterativno izvršavanje naredbe ili bloka naredbi, a upravlja se preko iteracijskog izraza (bilo koji Pythonov izraz koji se može iskoristiti kao argument).
DEF	Funkcija se definira. Primjer sintakse sa naredbom def: <pre>def funkcijsko_ime(parametri):     naredba(e)</pre>
BREAK	Naredba se koristi unutar tijela petlje. Kada se <i>break</i> izvrši, petlja završava. Ukoliko je petlja unutar druge petlje, <i>break</i> završava jedino petlju koja je pozicionirana najdalje unutra.
IF	Koristimo naredbu kada se niz naredbi treba izvršiti u slučaju ako uvjet nije zadovoljen. Ponekad je izvršavanje ovisno o nekoliko međusobno povezanih uvjeta. Naredba <i>if</i> koristi proširenje <i>elif</i> ili <i>else</i> zaključak.
WHILE	Naredba izvršava blok naredbi ovisno o ispunjenju upravljanog uvjeta. Prvo se izračunava uvjet petlje, ukoliko uvjet nije istinit, naredba <i>while</i> završava i niti jedna od naredbi u tijelu petlje se ne izvršava.
DEL	Naredba <i>del</i> ne briše objekte nego odvezuje njihove reference, jer brisanje objekata je implicirana posljedica sakupljanja smeća, kada više ne postoji nijedna referenca na taj objekt.

Python je podržan na Windows, Mac, Linux i Unix operativnom sustavu. Kod instalacije, Python paket instalira se u mapu c:\PythonXX, gdje XX predstavlja oznaku instalirane Python inačice. Ukoliko instaliramo Python 3.3, c:\Python33 je naziv mape u kojoj se nalazi instalirana verzija Pythona 3.3. Potrebno je postaviti i definirati Python putanju kako bi se Python mogao pokrenuti iz komandne linije. Trenutna inačica programa za Windows je Python 3.4.3.

Inačice Pythona do 25.02.2015.:

- Python 1.0 - Siječanj 1994.
- Python 2.0 - 16.10.2000.
- Python 3.0 - 03.12.2008.
  - Python 3.4 - 16.03.2014.
    - Python 3.4.1 - 19.05.2014.
    - Python 3.4.2 - 13.10.2014.
    - Python 3.4.3 - 25.02.2015.

Većina Python implementacije (uključujući i Cpython<sup>13</sup>) može funkcionirati kao komandna linija. U tom obliku rada upisuju se naredbe te je odmah vidljiv rezultat. Ljuska ( eng. *Shell* ) kao osnovni element Python-a zadužena je da interpretira naredbe. Značajke Python ljuske su *auto-completion*<sup>14</sup> (automatska nadopuna teksta) te zadržavanje sesije i sintakse.

Osim standardnih Python IDE-a, u razvoju aplikacije moguće je koristiti i *browser-based* IDE-i, Sage<sup>15</sup>, *browser-based* Integrirano razvojno okruženje i *hosting* okruženje PythonAnywhere.

---

<sup>13</sup> Cpython - izvorni interpreter Pythona

<sup>14</sup> Auto-completion - pomoć kod fizičkih nedostataka, pomoć za brže tipkanje

<sup>15</sup> Sage - matematički software koji pokriva velika područja matematike, algebre, teorije brojeva, numeričke matematike i kombinatorike. Namijenjeno za razvoj znanstvenih i matematičkih Python programa

#### 4.1. Povezivanje Python aplikacije sa bazom podataka

Za povezivanje sa bazom potrebno je instalirati MySQL *driver*. MySQL ima dobro dokumentirane module i ekstenzije te podršku od strane Python-a. Python instalacija dolazi sa SQLite<sup>16</sup>. Python može osim te relacijske baze koristiti i druge poznate baze podataka. Windows i Mac operativni sustavi koriste MySQLdb<sup>17</sup> dok Linux koristi standardni paket ( Python MySQLdb ).

Kod povezivanja sa bazom podataka, u settings.py potrebno je postaviti lozinku.

```
DATABASE = {  
    'default': {  
        'ENGINE': 'mysql.connector.django',  
        'NAME': 'rad',  
        'USER': 'root',  
        'PASSWORD': 'livona1',  
        'HOST': 'localhost',  
        'PORT': '3306'  
    }  
}
```

#### **Kod 1.** Povezivanja s bazom podataka:

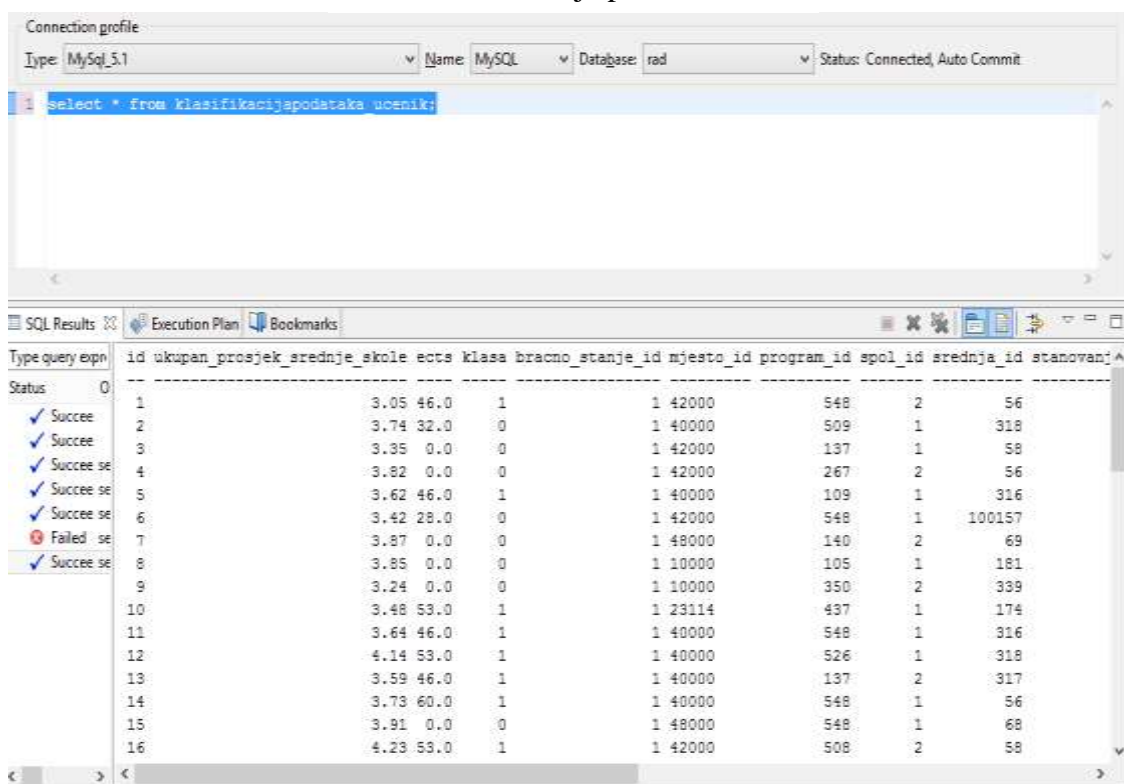
U Eclipse razvojnoj okolini koristi se *Database environment* za prikaz korisnika u bazi podataka. Na slici 3. prikazani su korisnici u bazi podataka koji se putem naredbe: `select * from klasifikacijapodataka_ucenik` selektiraju i prikazuju.

---

<sup>16</sup> SQLite - programska podrška koja omogućuje jednostavan sustav za upravljanje SQL bazama podataka.

<sup>17</sup> MySQLdb - sučelje MySQL-a za Python

Slika 3. Selektiranje podataka u bazi



## 4.2. Django i PyDev

Django je besplatna, open-source web razvojna okolina pisana u Python-u koja prati model - predložak - pregled (MTV<sup>18</sup>) arhitekturu. Osmišljen je s ciljem olakšanja stvaranja kompleksnih baza podataka na web stranicama. Razvijen je 2003. godine i nazvan je po gitaristu Django Reinhardt. U lipnju 2008 godine formiran je *Django Software Foundation* (DSF) te je Django postao dio DSF-a. Django se sastoji od alata koji olakšavaju razvoj i održavanje web aplikacija. Alati imaju čitav niz svojstva: komunikaciju s bazom podataka, autorizaciju, autentikaciju<sup>19</sup>, konfiguraciju URL-a i dr.

MVT arhitektura odvaja različite dijelove aplikacije; logiku primjenskog sustava<sup>20</sup> te prikaz i pristup podacima. U Django razvojnoj okolini kreira se zasebna datoteka pregleda (engl. *views*) prilikom kreiranja web aplikacija. U datoteci pregleda nalaze se funkcije pisane u programskom jeziku Python. Te funkcije upravljaju izvođenje upita kod web aplikacija. [3]

<sup>18</sup> MVT - Model Template View

<sup>19</sup> Autentikacija - proces određivanja identiteta nekog subjekta

<sup>20</sup> Primjenski sustav - Izvršni sustav

PyDev je dodatak za pisanje Python programa u Eclipse razvojnom okruženju. PyDev razvijen je u srpnju 2003. godine. Django integracija u PyDev-u radi preko 3 konfiguracije:

- *Django\_manage\_location* - string varijabla ukazuje na položaj projekta u `manage.py`,
- *Django\_settings\_module* - string varijabla mora sadržavati ime modula u projektu
- Projekt mora biti definiran kao Django projekt unutar PyDev-a.

#### 4.3. SciKit razvojni paket

SciKit learn je open source biblioteka za Python. Sadrži različite klase objekata za klasifikaciju, regresiju i ostale algoritme strojnog učenja (engl. *Machine Learning*). Biblioteka uključuje podršku za vektorske strojeve, logističke regresije i Naive Bayes-a. U SciKit paketu nalaze se NumPy i SciPy biblioteke koje su prilagođene Python okruženju. Biblioteke sadrže klase za razne standardne matematičke i numeričke funkcije.

NumPy (engl. *Numeric Python*) ima implementirane funkcije za manipulaciju velikih polja i matrica numeričkih podataka. Pri tome naglasak je na efikasnom računanju s vektorima, nizovima i matricama u višedimenzionalnim strukturama. Biblioteka (modul) je pisana u C-u i Fortanu te se služi BLAS bibliotekom<sup>21</sup>.

Naredba za učitavanje NumPy paketa preko modula: `from numpy import *`

SciPy (engl. *Scientific Python*) proširuje funkcionalnost Numpy-a sa značajnom zbirkom korisnih algoritama. Algoritmi kao što su algoritmi za minimizaciju, Fourierova transformaciju, regresiju i razne matematičke tehnike. Postoje različite vrste SciPy-a modula:

- `scipy.integrate` - integracija funkcija,
- `scipy.special` - specijalne funkcije,
- `scipy.optimize` - optimizacija,

---

<sup>21</sup> Basic Linear Algebra Subprograms je biblioteka rutina pisanih u FORTANU 77 koje obavljaju osnovne algebarske operacije

- `scipy.interpolate` - interpolacija,
- `scipy.linalg` - linearna algebra,
- `scipy.fftpack` - Fourierova transformacija,
- `scipy.sparse` - linearna algebra s rijetkim matricama,
- `scipy.stats` - statistika,
- `scipy.ndimage` - procesiranje slika.

```
Naredba za učitavanje SciPy paketa preko modula: from scipy import *
```

## 5. STROJNO UČENJE

Strojno učenje (engl. *Machine Learning*) je programiranje računala na način da optimizira nekakve kriterije uspješnosti na temelju podatkovnih primjera ili prethodnog iskustva. Temeljni pojmovi u strojnom učenju su indukcija<sup>22</sup> i generalizacija<sup>23</sup>. Na slici 4. prikazana su područja primjene strojnog učenja.

Strojno učenje ima tri usko vezane discipline:

- Data mining - rudarenje podataka,
- Probabilističko zaključivanje,
- Adaptivno procesiranje signala.

Kod strojnog učenja naglasak je na metodama koje garantiraju kvalitetu naučenih modela, karakteriziraju što je i pod kojim uvjetima naučeno te se skreće pozornost na automatiziranje korištenja i otkrivanja pravilnosti u podacima. Postoje različiti oblici strojnog učenja; nadzirano i nenadzirano učenje.

Nadzirano učenje (engl. *Supervised Learning*) je učenje gdje se koristi eksplicitna informacija o vrijednostima i primjerima njihove ciljne varijable. Cilj ovakvog učenja je razvoj modela kojim će se provoditi proces predikcije na još neviđenim primjerima. Na

---

<sup>22</sup> Indukcija - logička metoda, vrsta posrednog zaključka

<sup>23</sup> Generalizacija - jedna od osnovnih metoda spoznaje, postupak prelaska s općih na još općenitije pojmove



primjer, klasifikacija objekata, određivanje njihove pripadnosti nekoj od klasa ili predikcija. Nadzirano učenje najčešće se koristi u područjima:

- predikcije / klasifikacije,
- regresije.

U procesu klasifikacije nepoznatom objektu pridružujemo klasu pripadnosti na osnovu njegovih svojstava. Kod regresije, algoritam objektu pridružuje konstantnu vrijednost. Razlika je u tome je li ciljana varijabla, diskretna ili kontinuirana.

Nenadzirano učenje (engl. *Unsupervised learning*) je pristup gdje se pokušava riješiti određeni problem bez upotrebe prije zapisanih informacija ili formiranog modela učenja. Cilj takvog učenja je otkriti strukturu povezanost podataka u višedimenzionalnom okruženju. Nenadzirano učenje najčešće se koristi u tri područja:

- Grupiranje (engl. *clustering*),
- Smanjenje dimenzionalnosti (engl. *dimensionality reduction*),
- Otkrivanje novih vrijednosti ili vrijednosti koje odskaku (engl. *novelty/outlier detection*).

**Slika 4.** Strojno učenje



( Strojno učenje, Tomislav Šmuc ( 25.02.2011 ) )

## 5.1. Naive Bayes

Naive Bayes je metoda strojnog učenja čiji algoritmi pripadaju obitelji algoritama vjerojatnosti temeljenih na Bayes-ovom teoremu. Bayesov teorem poznat kao i Bayesian zaključak ostvaruje način izračunavanja vjerojatnosti hipoteze koja proizlazi iz pojedinih dijelova dokaza, kada su nam dane samo vjerojatnosti prema kojima dokazi proizlaze iz stvarnih posljedica (hipoteza).

Bayesov teorem tvrdi:

$$\Pr(A_i|B) = \frac{\Pr(B|A_i) \Pr(A_i)}{\sum_j \Pr(B|A_j) \Pr(A_j)} \quad (1)$$

Gdje su oznake:

- **Pr ( Ai | B )** - vjerojatnost da je Ai istinito kod danog dokaza B
- **Pr ( ai )** - vjerojatnost da je Ai općenito istinito
- **Pr ( B | Ai )** - vjerojatnost opažanja dokaza B kada je Ai istinito
- **N** - broj mogućih hipoteza

Naive Bayes je jednostavna metoda za klasificiranje podataka. Upotrebom metoda izrađuju se modeli koji dodjeljuju klase problematičnim labelama i predstavljeni su kao vektori određenih vrijednosti. Pri tome klase labela su odvojene iz konačnog skupa. Svi algoritmi Naive Bayes-ove metode pretpostavljaju vrijednosti određenih značajki neovisno o vrijednosti drugih značajki, s obzirom na vrijednost klase. Algoritam koristi metodu maksimalne vjerojatnosti tj. moguće je raditi sa Naive Bayes-ovim algoritmom bez prihvaćanja Bayes-ove vjerojatnosti ili bez korištenja Bayes-ovih metoda. Naive Bayes-ov algoritam koristi se kod:

- tekstualne klasifikacije,
- filtracije raznih spam-ova.

## 6. BAZA PODATAKA I KLASIFIKACIJSKI PODACI

MySQL Connector Python je standardni program za razvoj Python aplikacija koji podržava sve MySQL ekstenzije. MySQL Connector Python omogućava prijenos podataka između Pythona i MySQL-a koristeći protokole. Podržava veze koristeći TCP/IP socket<sup>24</sup>. MySQL Connector Python je API<sup>25</sup> implementiran koristeći Python, odnosno nije potrebno instalirati MySQL ili Python module, dovoljno je koristiti standardnu biblioteku. U razvoju aplikacije korištenja je verzija 1.2.3.

U organizaciji baze podataka korišten je reverzibilni inženjering. Reverzibilni inženjering proces je rekonstrukcije fizičkog i/ili modela objekt-veza iz postojeće sheme. Može se koristiti za osiguranje dokumentacije kod postojećih aplikacija, osiguranje bolje slike sheme pri razvoju aplikacije i osiguranje boljih smjernica prilikom izmjena sheme. Reverzibilni inženjering je izrazito koristan za praćenje veza stranog ključa (engl. *Foreign Key*) kroz shemu.

### 6.1. Django ORM

ORM (engl. *Object Relation Mapping*) je alat pomoću kojeg se šalju upiti o podacima te se manipuliraju podaci iz baze koristeći objektu paradigmu, odnosno korištenjem poveznica između objekata i tablica. U `models.py`, Django ORM učitava podatke te kreira tablice, veze i indekse. Također, omogućava migraciju podataka. Na temelju modela tijekom migracije, kreira se baza podataka ispunjena podacima iz CSV datoteke<sup>26</sup>.

```
class StrucnaSprema(models.Model):  
    id = models.IntegerField(primary_key=True)  
    sprema = models.CharField(max_length=50)
```

**Kod 2.** Primjer Stručne spreme sa id / sprema poljem:

U kodu 2. prikazana je izrada tablica u bazi podataka. Za primjer je uzeta tablica *Sprema*. *Id* i *Sprema* su polja modela. Svako polje je specificirano kao atribut klase, a

---

<sup>24</sup> TCP/IP socket - Network socket

<sup>25</sup> API - Aplikacijsko programsko sučelje

<sup>26</sup> CSV datoteka - Comma Separated Values

svaka klasa predstavlja red podataka u tablici baze podataka. Na temelju parametara; `primary_key=True` i `max_length=50`, kreira se tablica Sprema u `models.py`.

Struktura baze podataka može biti logička i fizička. Fizička struktura je implementacija u samoj bazi podataka dok se logička struktura realizira na sustavu za upravljanje relacijskim bazama podataka<sup>27</sup>. Struktura aplikacije logički se modelira putem ORM-a. Django ORM na temelju objektnog modela definiranog kroz `models.py` generira strukturu u relacijskoj bazi podataka. Baza podataka povezuje se pomoću `models.py`. Za prikaz podataka iz baze u HTML formatu koristi se `views.py`. Model dohvaća tražene vrijednosti iz baze podataka te ih prosljeđuje pogledu (`views.py`). `Views.py` filtrira te prikazuje tražene podatke.

```
def baza(request):
    context = RequestContext(request)
    ucenici = Ucenik.objects.all()
    return render_to_response('baza.html', {'ucenici' :
ucenici}, context)
```

**Kod 3.** Primjer primanja zahtjeva, dohvaćanja podataka iz baze i vraćanja vrijednosti na predložak

Kod 3. prikazuje primanje zahtjeva, dohvaćanje podataka iz baze te vraćanje vrijednosti na predložak. Varijabla `request` prima zahtjev. Dohvaćeni podaci iz baze podataka zapisuju se u varijabli `ucenici = Ucenik.objects.all()`, dok varijabla `response` vraća vrijednosti na predložak, odnosno na `baza.html`.

Django automatski generira tablice za potrebu administracije, migracije i autorizacije koristeći `django_admin_log`<sup>28</sup>, `content_type`<sup>29</sup>, `migrations`<sup>30</sup> i `session`<sup>31</sup>.

---

<sup>27</sup> RDBMS - Relation Database Management System

<sup>28</sup> `Django_admin_log` - koristi se za administraciju

<sup>29</sup> `Content_type` - prati modele u Django aplikaciji

<sup>30</sup> `Migrations` - prati sve promijene modela rađene u bazi podataka. Na primjer dodavanje polja, brisanje itd.

<sup>31</sup> `Session` - omogućuje pohranjivanje i dohvaćanje podataka

## 6.2. Atributi objekata / models.py

Atributi objekata osiguravaju metapodatke<sup>32</sup> koji se odnose na kvalitetu klasifikacijskih i položajnih podataka o objektu te metodi pohranjivanja i prikupljanja podataka. Oni se najčešće prikazuju u obliku tablice atributa za pojedine vrste objekata.

Model je jedan, konačan izvor informacija o podacima. Sadrži osnovna polja i podatke koji se spremaju te definira odnose i oblike podataka u bazama podataka. Općenito, svaki se model preslikava na jednu tablicu baze podataka. Svaki model je Python klasa koja pripada `django.db.models.Model`.

Svaki atribut modela predstavlja polje baze podataka te Django automatski generira API pristup bazi podataka. Python određuje varijable i metode koje su pridružene određenim tipovima podataka. Varijable imaju značenje redova u tablici dok su metode definirane kao relacija između varijabla.

## 7. PYTHON APLIKACIJA

Ova Python web aplikacija koristi SciKit razvojni paket. Aplikacija na osnovu vrijednosti atributa za studenta određuje vjerojatnost hoće li će student završiti studij. Algoritam provodi klasifikaciju objekta na temelju prijašnjih podataka o uspjehu bivših studenata i njihovih uspjeha. Rezultat klasifikacije uvijek je potrebno uzeti s određenom rezervom. Podaci studenata spremljeni su u bazi podataka koja je povezana s aplikacijom putem MySQL Connector-a. Korisnik u formi unosi tražene podatke o studentu: ime, prezime, spol, prebivalište, završenu srednju školu / struku / program u srednjoj školi, bračno stanje, status stanovanja, stručnu spremu majke, zaposlenje majke, stručnu spremu oca, zaposlenje oca te ukupan prosjek ocjena u srednjoj školi. Klikom na gumb „Izračunaj vrijednost“ algoritam na osnovu demografskih podataka određuje da li će student postići određeni broj ECTS-a odnosno hoće li završiti studij.

Klikom na gumb „Baza podataka“, upisani podaci iz forme spremaju se u bazu podataka te je u tablici prikazan novi upis forme. Također, u tabličnom prikazu vidljivi su podaci u bazi podataka koje algoritam upotrebljava u svojem izračunu.

---

<sup>32</sup> Metapodaci - podaci o podacima

Podaci studenata dohvaćaju se iz baze podataka putem modela, dok se pregled podataka ostvaruje se upotrebom pogleda (views.py).

Podaci studenata upisani su u CSV formatu. U datoteci, svakom atributu je dodijeljena oznaka. Svim demografskim podacima kao što su stručna sprema, bračno stanje, Zanimanje i tako dalje dodijeljena je šifra. Korišteni šifrnici u aplikaciji prikazani su u tablicama 1-5.

**Tablica 1.** Šifrnici zanimanja

Zanimanje	Opis zanimanja
1	članovi i čelnici zakonodavnih i upravnih tijela, te direktori
2	znanstvenici i najviši tehnički stručnjaci (diplomirani inženjeri i ostali visoki stručnjaci)
3	inženjeri, tehničari i srodni stručnjaci (uključuju zanimanja predradnika i poslovođa)
4	pisarnički i šalterski službenici
5	trgovinska i uslužna zanimanja
6	poljoprivredni, šumski, ribarski i radnici srodnih zanimanja
7	zanimanja pojedinačne izrade (uključuje održavanja, montažu, pojedinačnu proizvodnju i sl.)
8	rukovatelji proizvodnim strojevima, transportnim uređajima i vozilima, te sastavljači proizvoda
9	jednostavna zanimanja (uključujući čistače ulica, dostavljače, čistačice i sl.)
10	vojna zanimanja
11	umirovljenici i ostale osobe s osobnim prihodima (ne s plaćom)
12	uzdržavane osobe (kućanice i sl.)
13	privremeno nezaposleni

**Tablica 2.** Šifarnik za atribut bračno stanje

Bračno stanje	Opis statusa
1	slobodan
2	u braku

**Tablica 3.** Šifarnik za atribut status stanovanja

Status stanovanja	Opis statusa
1	kod roditelja
3	podstanar
4	u studentskom domu
5	u vlastitom domaćinstvu ili domaćinstvu supružnika
6	ostalo

**Tablica 4.** Šifarnik za attribute država rođenja, mjesta rođenja i poštanski broj

Država rođenja	Mjesto rođenja	Poštanski broj
HR	Varaždin	42000
HR	Čakovec	40000
HR	Koprivnica	48000
HR	Zagreb	10000
RS	Zrenjanin	23114
HR	Našice	31500
HR	Osijek	31000
HR	Split	21000
RU	Volgograd	400131

**Tablica 5.** Šifrarnik za atribut stručna sprema

Stručna sprema	Kratice
1	NKV
2	PKV
3	KV
4	VKV
5	NSS
6	SSS
7	VŠS
8	VSS
9	MR
10	DR

### 7.1. Moduli aplikacije

Python web aplikacija je modularna. Django skripta prilagođava instalaciju aplikacija to jest može sama instalirati sve potrebne aplikacije/ekstenzije ili se instalacija radi putem Python path-a.

```
INSTALLED_APPS = (  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'selectable',  
    'KlasifikacijaPodataka',  
)
```

#### **Kod 4.** instalirane ekstenzije u postavkama Python-a

U kodu 4. prikazane su instalirane ekstenzije u *settings.py* direktoriju. U Django datoteci sa postavkama nalaze se sve konfiguracije Django instalacije. Ekstenzije poput *django.contrib.admin*, *auth*, *contenttypes*, *sessions* i *messages* već su unaprijed



instalirane u postavkama. *Django.contrib.staticfiles* koristimo za učitavanje strateških podataka. Skuplja statičke podatke te ih sprema na lokaciju koju koristimo u radu aplikacije. *Django.contrib.selectable* je Django komponenta za Ajax upite na bazu podataka.

URL<sup>33</sup> konfiguracija za pokretanje web aplikacije zapisuje se u mapi *urls.py*. *Urls.py* konfigurira sve veze u projektu, odnosno veza je između url-a i Python funkcija (*views.py*).

```
urlpatterns = patterns('',
    # Examples:
    url(r'^$', 'KlasifikacijaPodataka.views.home_page',
name='home'),
    # url(r'^blog/', include('blog.urls')),
    url(r'^klasifikacija/',
'KlasifikacijaPodataka.views.klasifikacija',
name="klasifikacija"),
    url(r'^baza/', 'KlasifikacijaPodataka.views.baza',
name="baza"),
    url(r'^admin/', include(admin.site.urls)),
    url(r'^selectable/', include('selectable.urls')),
)
```

#### **Kod 5.** prikaz veze između URL-a i funkcija odnosno *views*

U kodu 5. prikazane su veze između URL-a i funkcija. Django putem Python modula traži varijable u *urlpatterns*. Zatim prolazi kroz svaki URL, zaustavlja se kod prvog URL-a koji odgovara traženom zahtjevu, uvozi ga i vraća pogledu (*views*). Naredba *r'* prenosi Pythonu da je veza „raw“ te da se ništa u navedenom *stringu* ne smije izostaviti.

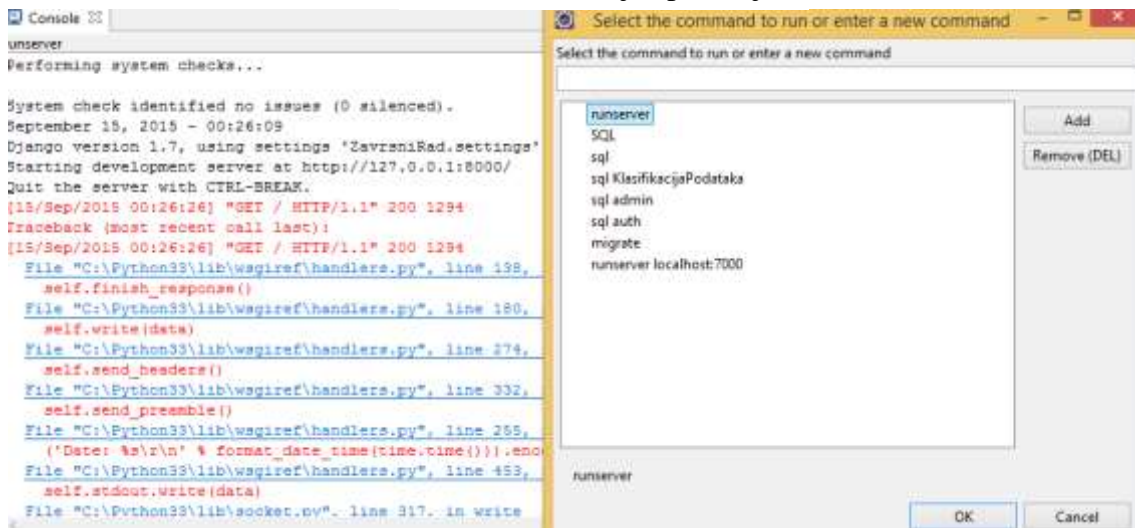
---

<sup>33</sup> URL - Uniform Resource Locator; usklađeni lokator sadržaja

## 8. TESTIRANJE APLIKACIJE

Za pokretanje aplikacije, u Django Custom Command-u potrebno je pokrenuti server te u konzoli pričekati provjeru sustava. Na slici 5. prikazan je proces pokretanja aplikacije. Aplikacija se pokreće upisivanjem *localhost:8000* u Internet preglednik.

Slika 5. Pokretanje aplikacije



Na slici 6. prikazana je početna stranica aplikacije. Aplikacija se pokreće klikom na gumb „Pokreni aplikaciju“.

Slika 2. Početna stranica



**Ova Python web aplikacija preko Naive Bayes-ovog algoritma  
izračunava vjerojatnost hoće li student završiti studij te prikazuje korisnike u  
bazi podataka**

Pokreni aplikaciju

Stranica klasifikacije prikazana je na slici 7. Na toj stranici nalazi se forma za upis korisničkih podataka. Korisnik može izabrati opciju „Izračunaj vrijednost“, „Baza podataka“ ili „Poništi“.

```
from KlasifikacijaPodataka.models import Ucenik
import django
import numpy as np
from sklearn.naive_bayes import MultinomialNB
```

#### **Kod 6.** postavljanje algoritma

U kodu 6. prikazano je početno postavljanje algoritma u direktoriju *algoritam.py*. *Sklearn* je Python modul u kojem se nalaze osnovne klase za strojno učenje. Također, povezuje razne algoritme i metode učenja s bibliotekama. Povezuje biblioteke kao što su NumPy i SciPy. *MultinomialNB* označava klasifikator pogodan u radu s diskretnim vrijednostima te klasifikator multinomnog modela.

U formi se upisuju podaci o novom studentu. Na temelju ulaznih podataka, algoritam određuje pripadnost tog novog studenta jednoj od klasa. Određuje pripada li student klasi onih studenata koji završavaju studij.

```
def klasifikacija(predict_value):
    django.setup()
    predict_value = [1, 40000, 316, 4, 109, 1, 1, 6, 5, 6, 5,
3.62]
    print("Klasa:", klasifikacija(predict_value))
```

#### **Kod 7.** Prikaz rada algoritma sa predviđenim vrijednostima

U kodu 7. prikazan je rad algoritma i postavljanje predviđenih vrijednosti u direktoriju *algoritam.py*. Naredbom *predict\_value* definiraju se vrijednosti podataka na temelju kojih će se vršiti klasifikacija. Vrijednosti *[1, 40000, 316, 4, 109, 1, 1, 6, 5, 6, 5, 3.62]* su oznake atributa spremljene u CSV formatu.

Klikom na gumb „Izračunaj vrijednost“, korisniku se ispisuje rezultat klasifikacije. Pritiskom na gumb „Poništi“, aplikacija briše podatke iz forme.

U formi je ugrađena validacija podataka. Za pravilnu klasifikaciju, korisnik treba upisati sve podatke studenta kojima se želi dodijeliti određena klasa.

```
import selectable.forms as selectable
spol = selectable.AutoCompleteSelectField(
    lookup_class=SpolLookup,
    label='Odaberi spol:',
    required=True,

    widget=selectable.AutoComboboxSelectWidget
)
```

### Kod 8. Provjera upisanih podataka u formi Spol

U kodu 8. provjerava se upis podataka u formi Spol. Provjera svih upisanih podataka u formi vrši se u direktoriju *forms.py*. Naredba *selectable.AutoCompleteSelectField* dopušta kreiranje nove stavke. Postavljanjem naredbe *required = True*, unos u navedenu formu je obavezan.

Slika 7. Forma za klasifikaciju

Klikom na gumb „Baza podataka“ uneseni podaci u formi spremaju se u bazu podataka. Nakon upisa, aplikacija u web pregledniku prikazuje cjelokupni sadržaj podataka u obliku tablice. U tabličnom prikazu slike 8. vidljivi su podaci korisnika unesenih u bazu podataka sa svojim kraticama.

```

if 'baza' in request.POST :
print("Ucenik je sačuvan")
    ucenik.save()
    return HttpResponseRedirect('baza')

```

### Kod 9. Dohvaćanje i spremanje podataka u bazu

U kodu 9. prikazano je dohvaćanje i spremanje podataka o studentu u bazu. Pri otvaranju forme za upis studenta uvjet `if 'baza' in request.POST` označava provjeru odabira baze za upis studenta. Nakon ispunjene forme, pritiskom na gumb „Baza podataka“ u konzoli se naredbom `print` ispisuje "Učenik je sačuvan". Zatim naredba `ucenik.save()` sprema podatke o studentu u bazu. Naredbom `return HttpResponseRedirect('baza')` otvara se web stranica s prikazom baze podataka i novim upisom studenta.

**Slika 8.** Tablični prikaz podataka o studentima

### Ispis studenata

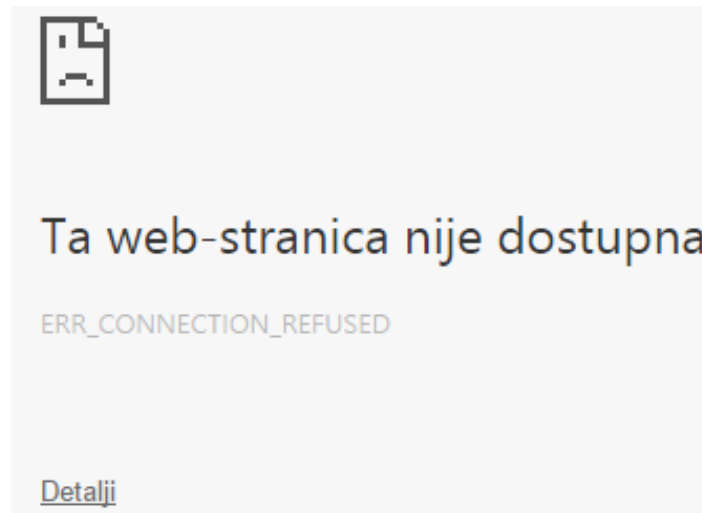
ID	spol	mjesto	srednja	struka	program	bracno stanje	stanovanje	SS majke	zap. majke	SS oca	zap. oca	prosjek
1	2	42000	56	36	548	1	1	6	5	6	9	3.05
2	1	40000	318	14	509	1	1	6	2	6	3	3.74
3	1	42000	58	6	137	1	1	6	7	6	5	3.35
4	2	42000	56	17	267	1	1	6	5	6	3	3.82
54	1	42000	56	21	31	1	1	5	12	5	7	3.83
55	2	42000	56	21	31	1	1	6	9	6	3	3.82
56	2	42000	58	7	148	1	1	6	3	6	1	3.55
57	1	42000	56	21	31	1	1	6	6	6	11	3.81
58	2	42000	56	17	267	1	1	6	7	6	7	4.59
59	1	42000	56	21	537	1	1	5	13	5	5	3.93
60	1	10000	34	36	548	1	3	3	2	2	2	4.0
61	2	40000	34	36	548	1	3	2	2	2	2	5.0
62	2	31000	34	36	548	2	3	4	2	4	3	5.0

Početna    Klasifikacija

Na svakoj stranici aplikacije nalaze se navigacijske tipke. Pritiskom na navigacijsku tipku „Početna“, aplikacija korisnika vraća na početnu stranicu. Pritiskom na tipku „Klasifikacija“, aplikacija korisnika vraća na stranicu klasifikacije.

Ukoliko se prilikom pokretanja aplikacije ne pokrene server, Internet preglednik prikaže pogrešku u radu aplikacije. Na slici 9. prikazana je poruka o takvoj situaciji.

**Slika 9.** Greška u Internet pregledniku



## 9. ZAKLJUČAK

Ovim završnim radom prikazana je upotreba Python programskog jezika za razvoj web aplikacije. Prikazan je postupak razvoja aplikacije, osnovni moduli i potrebne biblioteke.

U razvoju aplikacije korištena je Eclipse razvojna okolina, SciKit razvojni paket, PyDev alat, Django ORM alat te MySQL Connector Python. Uspješno je provedena integracija različitih programskih jezika i web okruženja.

Nadogradnja i poboljšanje aplikacije moguće je s podacima studenata koji su već prošli određenu klasifikaciju te su im dodijeljene vrijednosti za tražene podatke. Aplikacija ima velikih potencijala te se može primijeniti na raznim visokim učilištima.

Python uz svoju jednostavnost ipak ima elemente izrazito snažnog programskog jezika tako da se Python može usporediti sa malim švicarskim nožićem među programskim jezicima.

Izrada ove Python web aplikacije za mene je bila posve novo iskustvo, ponajviše jer sam se prvi put susrela sa spomenutim jezikom.

## 10.LITERATURA

**Scott Shell M.** (2014). An introduction to Numpy and Scipy

**Essert M.; Vazle I.** (2007). Python osnove. Osijek

**Scipy** - <http://web.math.pmf.unizg.hr/nastava/matsoft/Scipy.slides.html#/>

(20.06.2015)

**Numpy** - <http://web.math.pmf.unizg.hr/nastava/matsoft/Numpy.slides.html#/>

(20.06.2015)

**Nadzirano učenje** - [http://www.fer.unizg.hr/\\_download/repository/su-](http://www.fer.unizg.hr/_download/repository/su-2011-biljeska-1.pdf)

2011-biljeska-1.pdf (22.06.2015)

**Strojno učenje, grupiranje** - [http://www.fer.unizg.hr/\\_download/repository/SU](http://www.fer.unizg.hr/_download/repository/SU13-Grupiranje.pdf)

13-Grupiranje.pdf (22.06.2015)

**Osnove geoinformatike** - [http://www2.geof.unizg.hr/~barkovic/templates/doku](http://www2.geof.unizg.hr/~barkovic/templates/dokument/osnove_geoinformatike/XII_predavanje_POLOZAJ_OBJEKTA_OPISAN_AT_RIBUTOM.pdf)

me nt/osnove\_geoinformatike/XII\_predavanje\_POLOZAJ\_OBJEKTA\_OPISAN\_AT

RIBUTOM.pdf (30.06.2015)

**Models** - <https://docs.djangoproject.com/en/1.8/topics/db/models/> (02.07.2015)

**Sigurnost Django web razvojnog okruženja** - [http://os2.zemris.fer.hr/ns/websec](http://os2.zemris.fer.hr/ns/websec/2008_mihaljevic/)

/2008\_mihaljevic/ (19.09.2015)

**Django** - [http://www.pydev.org/manual\\_adv\\_django.html](http://www.pydev.org/manual_adv_django.html) (05.07.2015)

[1] Guido Personal Home Page - <https://www.python.org/~guido/> (18.09.2015)

[2] Python Success Stories - [https://www.python.org/about/](https://www.python.org/about/success/)

success/ (18.09.2015)

[3] Sigurnost Django web razvojnog okruženja - [https://www.python.org/about/](https://www.python.org/about/success/)

success/ (19.09.2015)

(1) Bayesov teorem - [http://autopoiesis.foi.hr/wiki.php?edit=yes&name=Ekspertni](http://autopoiesis.foi.hr/wiki.php?edit=yes&name=Ekspertni+sustavi&page=Bayesovo+pravilo)

+sustavi&page=Bayesovo+pravilo (19.09.2015)