

Integracija podataka putem razvojnog okružja Jupyter Notebook

Vukadinović, Andrej

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:211:236630>

Rights / Prava: [Attribution-NonCommercial-NoDerivs 3.0 Unported / Imenovanje-Nekomercijalno-Bez prerađivanja 3.0](#)

Download date / Datum preuzimanja: **2024-11-16**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Andrej Vukadinović

**INTEGRACIJA PODATAKA PUTEM
RAZVOJNOG OKRUŽJA JUPYTER
NOTEBOOK
ZAVRŠNI RAD**

Varaždin, 2020.
SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Andrej Vukadinović

Matični broj: 35918/07–R

Studij: Primjena informacijske tehnologije u poslovanju

INTEGRACIJA PODATAKA PUTEM RAZVOJNOG
OKRUŽJA JUPYTER NOTEBOOK
ZAVRŠNI RAD

Mentor:

prof. dr. sc. Schatten Markus

Varaždin, lipanj 2020.

Andrej Vukadinović

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Glavni cilj ovoga rada je približiti način integracije podataka putem razvojnog okruženja Jupyter Notebook. Ukratko ćemo objasniti sam Jupyter Notebook, čemu služi i koje su njegove mogućnosti. Potom ćemo korak po korak objasniti načine instalacije Jupyter Notebooka. Opisati ćemo način rada u okruženju Jupyter notebook povezati ga sa programskim alatima Pandas i SQLAlchemy te korak po korak prikazati način integracije podataka u navedenom alatu preko zamišljenog primjera baza podataka osnovnih škola koji smo sami izradili.

Ključne riječi: Jupyter Notebook, Anaconda, Python, Analitika velikih skupova podataka, Pandas, SQLAlchemy, Podatkovna znanost,

1. Uvod	v
2. Pregled korištenih tehnologija	1
2.1. Jupyter Notebook.....	1
2.1.1. Što je Jupyter notebook?	1
2.1.2. Povijest Jupyter notebook-a	1
2.1.3. Kako koristiti Jupyter Notebook?.....	2
2.1.3.1. Instalacija	2
2.1.4. Korištenje Jupyter Notebooka	5
2.1.4.1. Markdown	7
2.1.4.2. Kod.....	9
2.2. Pandas.....	9
2.2.1. Što je Pandas	9
2.2.2. povijest	10
2.2.3. Instalacija.....	10
2.2.4. Mogućnosti.....	10
2.3. SQLAlchemy(1-2 stranica)	11
Instaliranje SQLAlchemy-ja.....	12
2.3.1. Funkcionalnost SQLAlchemy alata	12
2.4. SQLite i MySQL	13
2.4.1. Sqlite.....	13
2.4.2. MySQL(MariaDB)	14
2.5. XAMPP	15
2.5.1. Instalacija XAMPP-a.....	15
2.5.2. Pokretanje servera baze podataka.....	15
3. Praktični rad.....	16
3.1. Rad sa MySQLom	16
3.1.1. Spajanje na server.....	16
3.1.2. Kreiranje baze podataka	17
3.1.3. Ispis baza podataka.....	17
3.1.4. Kreiranje tablice	18
3.1.5. Popunjavanje tablice.....	19
Ispisivanje vrijednosti tablice „rezultati“	19
3.2. Rad sa SQLite.....	20
3.2.1. Kreiranje baze.....	20
3.2.2. Stvaranje tablice	20
3.2.3. Popunjavanje tablice i ispisivanje vrijednosti	22

3.3. SQLAlchemy i Pandas	22
3.3.1. Čitanje tablica - pandas	23
3.3.2. Pretvaranje SQLite baze podataka u podatkovni okvir	23
3.3.3. Prikaz određenih polja podatkovnog okvira.....	24
3.3.4. Mjenjanje imena stupaca podatkovnog okvira.....	24
3.3.5. Spajanje tablica – konkatencija	25
3.3.6. Računanje prosjeka po predmetu (stupcu) – mean.....	26
3.3.7. Računanje prosjeka učenika (retci)	26
3.3.8. Prosjek ocjena učenika - prosjek svih učenika.....	27
3.4. Matplotlib	28
4. Kritički prikaz.....	29
5. Zaključak	29
Popis literature.....	31
Popis slika	33

1. Uvod

Navedenu temu sam odabrao zbog interesa u polje velikih podataka i analize podataka i smatrao sam da bi mi ovakva tema ponudila dovoljno motivacije da uđem u materiju i programski kod navedenog područja. Također ne možemo negirati da je podatkovna znanost je u rastu, i mnoštvo se programa koristi za obradu, prikupljanje i prikazivanje podataka, međutim u znanstvenoj zajednici Jupyter Notebook ima reputaciju kao jedna od najpoznatijih i najkvalitetnijih aplikacija za prikazivanje i obradu podataka. U slijedećem radu ćemo pogledati zašto je Jupyter Notebook toliko cijenjen, koje su njegove prednosti kada ga stavimo pored drugih programa. Također ćemo proći kroz programske biblioteke Pandas i SQLAlchemy te ih koristiti u programskom okruženju Jupyter Notebook-a.

2. Pregled korištenih tehnologija

2.1. Jupyter Notebook

2.1.1.Što je Jupyter notebook?

Jupyter notebook je web aplikacija tipa otvorenog koda koja služi za stvaranje i podjelu dokumenata koji sadrže „živi kod“, jednadžbe, vizualizacije i tekst HTML, vizualizacije, LaTeX zapis, te MIME tipove podataka. Kao program se često koristi u prezentacijske svrhe zbog svojega dizajna koji djeli tekst, kod i vizualizacije u različite ćelije te im tako daje preglednost. (Driscoll 2017) Glavna prednost Jupyter Notebooka je njegova mogućnost da generira dokumente sa visokom razinom interaktivnosti. Čitatelj navedenih dokumenata može dobiti primjer koda. Taj kod je moguće pokrenuti u istom dokumentu te vidjeti njegov output. Ispod njega može naći tekst sa opisom navedenog koda, njegovu funkciju itd. Također uz LaTeX zapis moguće je u dokument stavljati matematičke jednadžbe. Korištenje MIME podataka u dokument je moguće ubaciti video zapise, slike i audio. Jupyter Notebook je u smislu programskih jezika agnostičan što znači da nije ovisan o jednom programskom jeziku već može koristiti nekoliko različitih u jednom dokumentu. Trenutačno postoji 49 jezgri (eng. kernela) koji su kompatibilni sa Jupyter Notebook-om te podržavaju mnoge programske jezike kao što su Julia, Python, R i Haskell. (Jupyter Team,2015)

Dakle Jupyter Notebook proširuje pristup interaktivnom programskom kodu baziranom na konzoli tako što pruža web aplikaciju koja obuhvaća prikaz cijelog aspekta programskoga procesa (whole computation process): razvoj, dokumentacija i izvršenje koda, te izvještavanje o rezultatima. (Jupyter Team,2015)

2.1.2.Povijest Jupyter notebook-a

Jupyter Notebook je nastao od strane neprofitne organizacije zvane Project Jupyter, te je zapravo počeo kao spinoff projekta zvanog IPython. Naime nakon što je IPython zbog svojeg razvoja počeo dodavati funkcionalnosti koji nisu morale biti specifične samo za Python, odlučili su ostvoriti novi projekt nazvan Project Jupyter kako bi te navedene funkcionalnosti bile dostupne preko više programskih jezika. IPython se i dalje može koristiti kao kernel Jupytera i kao interpreter u programskom jeziku Python, međutim druge funkcionalnosti kao što su mogućnost stvaranja Notebook dokumenata, su se prebacili pod novo ime Jupytera. Jupyter notebook je najavljen na SciPy-ovoj stranici 2014 godine. (Driscoll 2017)

2015 godine Jupyter Notebook je najavio kako će se format Jupyter Notebooka .ipynb moći pokretati izravno na stranici GitHub-a.(Shirokov 2015)

U sljedećim godinama Project Jupyter dodavao je nove funkcionalnosti na Jupyter Notebook te povećavao broj kompaktilnih programskih jezika. (Driscoll 2017)

2.1.3.Kako koristiti Jupyter Notebook?

2.1.3.1. Instalacija

Mi kao kernel Jupyter Notebooka koristimo Python. Verzija Pythona korištena tijekom pisanja ovoga rada je Python 3.8.5. Jupyter Notebook nije uključen u Python, stoga ako ga želimo koristiti moramo ga instalirati. Instalirati Jupyter Notebook možemo na dva različita načina. Jedan je koristeći pip u Pythonu, drugi način je instaliranje u sklopu Anaconda distribucije (Driscoll 2017)

1.Instaliranje sa Pip-om

Pip je system za upravljanje paketima koji je de facto standard za Python. Sistem za upravljanje paketima je skup softverskih alata koji automatiziraju i olakšavaju process instaliranja, nadogradnje, konfiguriranja i brisanja programa. Pip dolazi već instaliran sa Pythonom od Python verzije 2.7.9 na dalje. (debian.org, n.d.)

Kako bi instalirali Jupyter Notebook pomoću Pip-a trebamo znati lokaciju mapu u koji smo instalirali Python te navigirati u podmapu u kojoj se nalazi Pip.exe izvršna datoteka.

U mojem slučaju adresa mape je:

```
"C:\Users\Andrej\AppData\Local\Programs\Python\Python38-32\Scripts"
```

Nakon što smo našli lokaciju potrebne mape otvaramo cmd.exe (windowsov naredbeni redak) u administratorskom načinu rada kako bi bili sigurni da nećemo imati problema sa odobrenjima koji bi mogli omesti našu instalaciju.

Zatim upisujemo sljedeću naredbu:

```
>cd [Ovdje zaljepimo adresu mape u kojoj se nalazi Pip]
```

Primjer:

```
">cd C:\Users\Andrej\AppData\Local\Programs\Python\Python38-32\Scripts"
```

Rezultat bi trebao biti sličan kao na slici xxxxxxxxx dolje

```
C:\Users\Andrej\AppData\Local\Programs\Python\Python38-32\Scripts>dir
Volume in drive C has no label.
Volume Serial Number is 9EBF-5E53

Directory of C:\Users\Andrej\AppData\Local\Programs\Python\Python38-32\Scripts

31/08/2020  19:21    <DIR>          .
31/08/2020  19:21    <DIR>          ..
31/08/2020  19:21                97,172 easy_install-3.8.exe
31/08/2020  19:21                97,172 easy_install.exe
31/08/2020  19:21                97,163 pip.exe
31/08/2020  19:21                97,163 pip3.8.exe
31/08/2020  19:21                97,163 pip3.exe
                5 File(s)      485,833 bytes
                2 Dir(s)    362,179,334,144 bytes free

C:\Users\Andrej\AppData\Local\Programs\Python\Python38-32\Scripts>
```

Slika 1 Instaliranje Jupyter Notebook-a sa PIP-om

Na slici xxxxxxxx (gore) važno je primjetiti kako za vrijeme pisanja ovoga rada, verzija pip-a koja se koristi je pip3. U slučaju da instalirate Jupyter Notebook na ovaj način u budućnosti bitno je provjeriti verziju pip-a, ako ona ne odgovara bitno je prilagoditi sljedeći korak novoj verziji tako da upišete odgovarajuće ime izvršne datoteke.

Kako bi završili instalaciju Jupyter Notebooka preko pip-a upišemo:

“pip3 install jupyter” te pritisnemo enter.

Pip će sada instalirati Jupyter Notebook na vase računalo. Kako bi olakšali korištenje programa u prethodno otvorenoj mapi možemo naći novu izvršnu datoteku zvanu “jupyter.exe” te stvoriti njezin prečac na radnoj površini kako se ne bi morali konstantno vraćati u tu mapu. (Driscoll 2017)

2.Instaliranje u sklopu Anaconde

Drugi način na koji možemo instalirati Jupyter je sa Anacondom. Kako nam Anaconda dodaje dodatne programske pakete u ovom radu smo se odlučili za taj način. Anaconda je program sa kodom otvorenog tipa koji je verzija programa Python i R. Dolazi sa mnoštvom različitih alata namjenjenim za podatkovnu znanost, strojno učenje, obrađivanje velikih skupova podataka I tako dalje. . (Anaconda.org 2021)

Glavna zadaća Anaconde je olakšavanje upravljanjem programskim paketima, no za razliku od Pip-a, Anaconda također ima grafičko sučelje koje olakšava pristup navedenim programskim paketima, kao što je u našem slučaju Jupyter Notebook (Anaconda.org 2021)

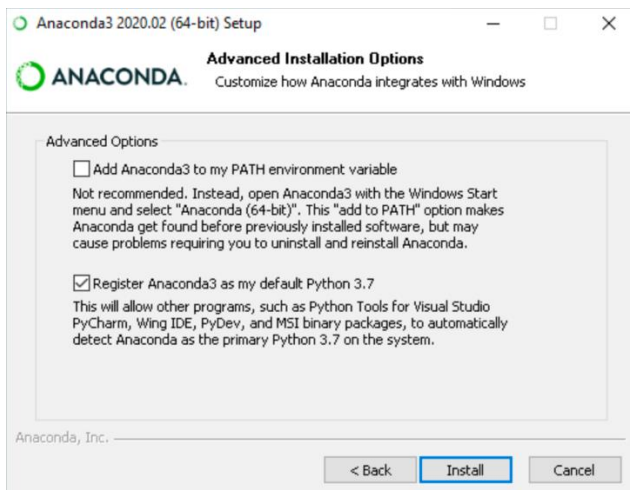
Kako Anaconda već u sebi sadži Jupyter Notebook trebamo samo instalirati Anakonu što je vrlo jednostavno. Također jedna od velikih prednosti Anaconde je to što se instalira u „Home“ directory, zbog toga ne trebamo Administratorska prava kako bi je instalirali. Te se samo uklanjanje Anakonde uvelike olakšava.

1.Kako bi instalirali Anacondu prvo trebamo skinuti instalaciju. Instalaciju možemo naći na njihovoj službenoj stranici na sljedećoj poveznici:

<https://www.anaconda.com/products/individual>

Bitno je spomenuti da ćemo mi koristiti individualnu verziju jer je besplatna. Također je bitno napomenuti da je Anaconda dostupna za Windows, MacOS I Linux, stoga je potrebno skinuti odgovarajuću instalaciju. (Anaconda.org 2021)

2.Kada skinemo instalacijsku datoteku, pokrenemo je, te pratimo instalacijski process. Ukoliko nemamo prijašnjih instalacija Pythona na računalu opciju “Add Anaconda3 to my PATH environment variable” ne bi trebali označavati. Inače može doći do problema koji će zahtjevati ponovnu instalaciju. . (Anaconda.org 2021)



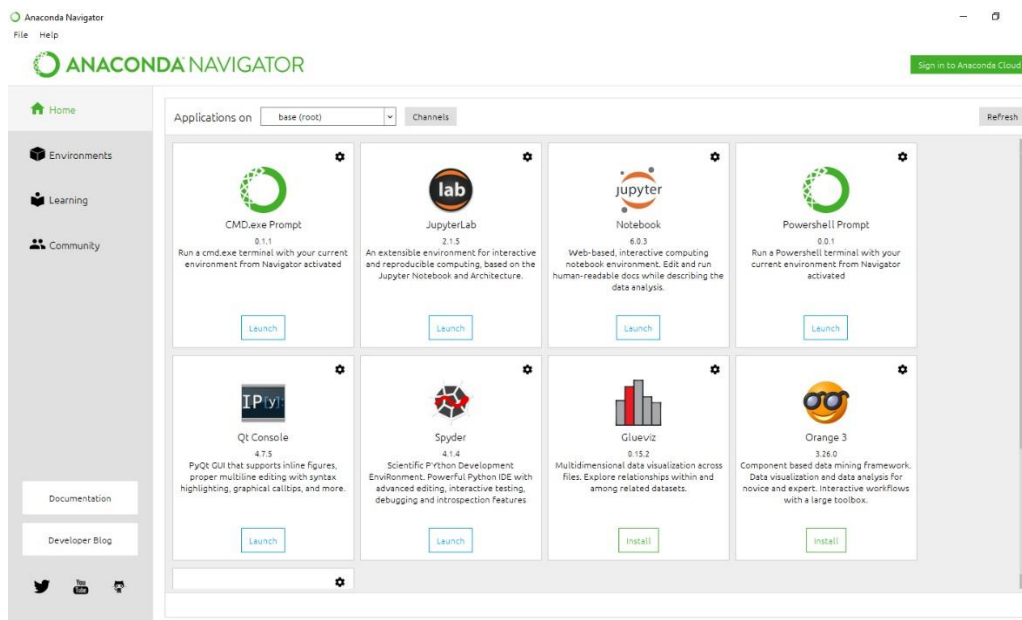
Slika 2 Instalacija Anaconda

2.1.4. Korištenje Jupyter Notebooka

Pošto smo našu verziju Jupyter Notebooka instalirali u sklopu Anaconde, najlakši način za korištenje je preko Anaconda Navigatora, tj. Anacondinog grafičkog korisničkog sučelja.

Tako da kliknemo na gumb „launch“ ispod ikone Jupyter Notebooka

Nakon toga nam se otvara prozor aplikacije u pregledniku našeg odabira.

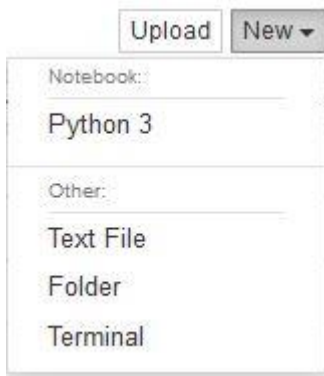


Slika 3 Anaconda Navigator

U podprozoru „Files“ možemo pretraživati mape na našem računalu, izraditi novu datoteku Jupyter Notebooka ili napraviti novu mapu u koju možemo spremati svoje projekte.

Novu mapu ili datoteku možemo izraditi tako da klikom na gumb „New“ te ako želimo izraditi datoteku kliknemo na Python 3, a ako želimo izraditi tekstualnu datoteku, novu mapu ili pak otvoriti novu instancu terminala kliknemo na pripadajuće izbore. . (Anaconda.org 2021)

Ako Jupyter Notebook pokrećemo na Web Serveru tipkom „Upload“ možemo „uploadati“ podatke sa računala na server. (Driscoll 2017)

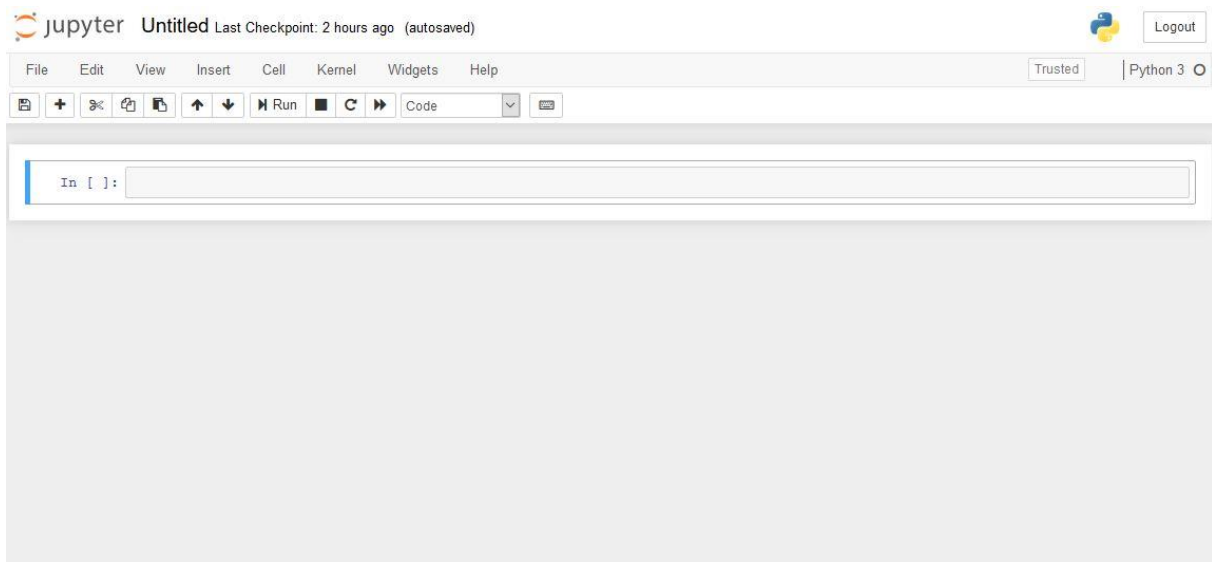


Slika 4 Tipovi ćelija u Jupyter Notebook-u



Slika 5 Jupyter browser

U potprozoru „running“ možemo naći sve trenutačno pokrenute Jupyter Notebook datoteke i terminale. Nakon što smo odabrali koju datoteku Jupyterove Bilježnice ćemo odabrati otvara nam se Notebook Editor.



Slika 6 Izgled Jupyter Notebooka

Kao što smo rekli Jupyter Notebook može prikazati različite vrste podataka. Podatci se upisuju u ćelije. Jedna ćelija podržava jedan tip podatka. Npr. Ako u jednoj ćeliji počnemo pisati kod, u istoj ćeliji ne možemo zapisati Markdown jezik. Kako bi to pojasnili prikazati ćemo neke te dati primjere na koji se mogu koristiti. (Driscoll 2017)

2.1.4.1. Markdown

Markdown je popularan i jednostavan markap programski jezik. Koristeći Markdown možemo napraviti tekstualne datoteke te ih po potrebi urediti tako da su ugodne oku „
Markdown <http://www.aaronsw.com/weblog/001189>“

Neke od opcija koje Markdown nudi su: mjenjanje boje teksta, oldanje teksta, stvaranje zaglavlja, stvaranje zaglavlja, stvaranje poveznica, prikaz medijskih zapisa kao što su video, slika i zvuk, LaTeX zapis matematičkih formula i tako dalje.

```

# Zaglavlje - liste
Primjer liste:
* jedan
* Dva
* tri
  * 3.5
    * 3.551
* četiri

Primjer numerirane liste:
1. Čovjek
2. Slon
3. Pas
  1. Terijer
  2. Dalmatinac
4. Mačka

## Zaglavlje druge razine - poveznice i medija
[Jupyter Notebook](https://jupyter.org/)

<img src = "https://jupyter.org/assets/hublogo.svg" height = 200 width = 600>

### Zaglavlje treće razine - obrada teksta
<font color = "green"> Ovo je zeleni tekst </font>
<font weight = bold> Ovo je boldani tekst </font>

# LaTeX zapis jednadžbi

$$e^{i\pi} + 1 = 0$$



$$s = (b+c/d)$$


```

Slika 7 Izgled Jupyter Notebook Markdown ćelija

Kako bi naše ćelije u Jupyter Notebooku prepoznale da se radi o Markdown jeziku, moramo promjeniti tip ćelije u Markdown tip.

To radimo tako da u alatnoj traci na gornjem djelu programa kliknemo na „Cell“ nakon čega nam se prikazuje padajući izbornik. Nakon toga kliknemo na „Cell Type“, te „Markdown“ (Driscoll 2017).

Zaglavlje - liste

Primjer liste:

- jedan
- Dva
- tri
 - 3.5
 - 3.551
- četiri

Primjer numerirane liste:

1. Čovjek
2. Ston
3. Pas
 - A. Terijer
 - B. Dalmatinac
4. Mačka

Zaglavlje druge razine - poveznice i medija

[Jupyter Notebook](#)



Zaglavlje treće razine - obrada teksta

Ovo je zeleni tekst Ovo je boldani tekst

LaTeX zapis jednadžbi

$e^x + 1 = 0$

$a = (b + cd)$

Slika 8 Jupyter Notebook sa slikom unutar ćelije

2.1.4.2. Kod

Kodiranje u Jupyter Notebooku je isto kao i kodiranje u bilo kojem drugom programskom jeziku, ali jedna od razlika je u tome da djelove koda možemo pokrenuti neovisno o drugim djelovima koda što daje veliku fleksibilnost prilikom isprobavanja raznih kodova

2.2. Pandas

2.2.1.Što je Pandas

Pandas je softverska biblioteka koja je napisana za programski jezik Python te služi za manipulaciju i analizu podataka. Pandas se specijalizira u strukture podataka i operacije za manipulaciju numeričkim tablicama i vremenskim serijama. (The pandas development team, 2008-2021)

Misija Pandas tima glasi:

„Pandas ima za cilj biti „temeljni gradivni“ element za praktičnu analizu podataka iz stvarnog svijeta u programskom jeziku Python. Uz to, širi je cilj postati najmoćniji i najfleksibilniji

alat za analizu / manipulaciju podacima otvorenog koda dostupan na bilo kojem jeziku.“ (The pandas development team, 2008-2021)

2.2.2.povijest

AQR Capital započeo je sa razvojem Pandasa 2008 godine, te je program do kraja 2009. godine postao programom otvorenog koda 2012 je godine objavljeno je prvo izdanje knjige „Python za analizu podataka“ te je ono preporučavalo upotrebu Pandasa. 2015 godine pandas postaje projekt sponzoriran od strane NumFOCUS-a (The pandas development team, 2008-2021)

2.2.3.Instalacija

Instaliranje Pandasa možemo obaviti na nekoliko različitih načina. Mi ćemo objasniti 2 najčešće korištena.

1. Instaliranje pomoću Pipa

Instaliranje Pandasa sa Pipom je jako slično kao instaliranje Jupyter Notebooka u prethodnom primjeru međutim postoji i lakša varijanta. Otvorimo naš Python te te samo upišemo slijedeće naredbe: `python -m pip install pandas`

Nakon toga Pip će instalirati najnoviju verziju Pythona na naše računalo

2. Instaliranje u sklopu Anaconde

Kao što smo naveli u prethodnom primjeru Anaconda se sastoji od mnoštva različitih programskih paketa za Python. Jedan od njih je Pandas. U prethodnom primjeru smo objasnili kako se instalira Anaconda. Na službenoj Pandas stranici je predloženo za početnike da instaliraju Pandas u sklopu Anaconde jer je daleko jednostavnije. (Anaconda.org 2021)

Za instaliranje također Pandasa također možemo koristiti Miniconda instaler koji je minimalna verzija Conda paketa koji je iznimno manji, tako da možemo uštedjeti na prostoru i podatkovnom prometu. (The pandas development team, 2008-2021)

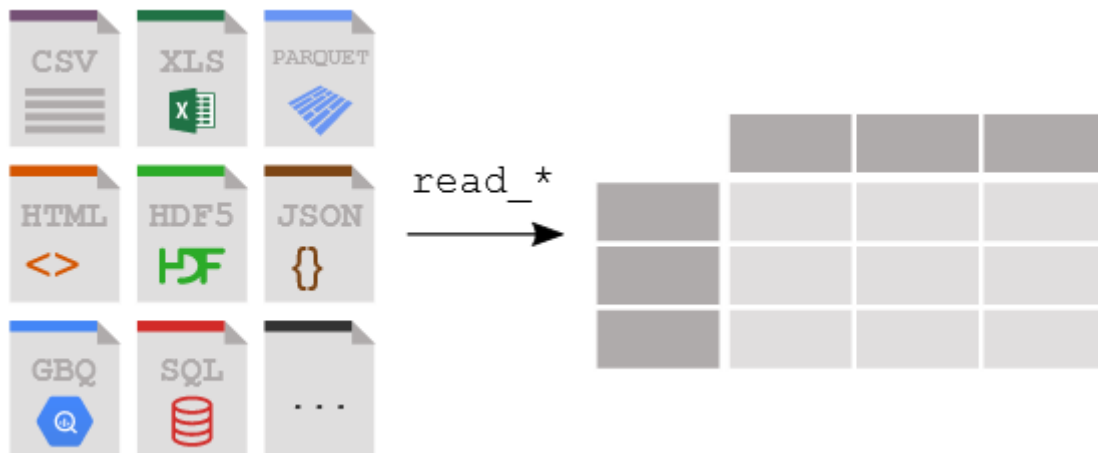
2.2.4.Mogućnosti

Ako tek počinjemo sa programskim pakrtom Pandas, najbolje rešenje bi bilo koristiti

Kao što smo već naveli glavna funkcionalnost specijalizira u strukture podataka i operacije za manipulaciju numeričkim tablicama i vremenskim serijama, a za to koristi tzv. Data frameove, tj dvodimenzionalne podatkovne strukture koje mogu u sebi sadržavati različite

tipove podataka kao što su integer, znakovi, decimalne brojeve, podatci za kategoriziranje (The pandas development team, 2008-2021)

Pandas navedene dataframeove može importirati iz raznih formata kao što su csv, excel, sql, json, parquet itd.



Slika 9 Pandas podržava mnogo različitih vrsta podataka

2.3. SQLAlchemy(1-2 stranica)

SQLAlchemy je Python SQL paket alata, napravljen sa kodom otvorenog tipa od strane Mike Bayera, koji programerima aplikacija daje punu snagu i fleksibilnost SQL -a. (Copeland 2008)

SQLAlchemy nudi veliki izbor funkcija koje su dizajnirane za učinkovit pristup bazama podataka kao što su Oracle, DB2, MySQL, PostgreSQL, i SQLite. SQLAlchemy služi za implementaciju objektno relacijskog mapiranja (eng. object relational mapping), ali se također SQLAlchemy može koristiti kao SQL jezik neovisan o poslužitelju baze podataka (SQLAlchemy, 2021) Glavna prednost SQLAlchemyja je to što pojednostavnije rad pružajući lak pristup funkcijama za spajanje na različite baze podataka, koristeći jednostavnu i shvatljivu sintaksu Python programskog jezika što objašnjava njegovu popularnost među podatkovnim znanstvenicima. . (Copeland 2008)

Instaliranje SQLAlchemy-ja

SQLAlchemy je iznimno lako za instalirati. Naime kao i sa Pandasom koristit ćemo dvije metode:

1. Instaliranje pomoću Pipa

Otvorimo naš Python te te samo upišemo slijedeće naredbe: `python -m pip install sqlalchemy` Nakon nekoliko sekundi, navedeni paket alata bi se trebao skinuti i instalirati na računalo. (SQLAlchemy 2021)

2. Instaliranje u sklopu Anaconde

SQLAlchemy, kao i mnogi drugi alati za rukovanje sa podacima se već nalaze u sklopu Anaconda distribucije. Tako da instaliranjem Anaconde dobijamo i SQLAlchemy, te ako ga želimo koristiti samo ga trebamo importirati (Anaconda.org 2021)

2.3.1. Funkcionalnost SQLAlchemy alata

Glavna prednost SQLAlchemy alata je pojednostavljenje korištenja baza podataka. Neka sučelja niže razine za Python kao što je DB-API koriste se dugom i kompleksnom sintaksom koja nakon nekoliko ponavljanja može postati repetitivna i zamorna, te također u nekim slučajevima otvara mogućnost SQL injection napada koji može iskoristiti greške u sistemu, te loše napisanu sintaksu kako bi došao do osjetljivih podataka iz baze. Također zbog svoje mogućnosti da se lako spaja na više DBMS-a, on uvijek ostavlja mogućnost prebacivanja na drugi DBMS promjenom tek nekoliko linija koda. Još jedan od ciljeva SQLAlchemy-ja je premoštavanje neusklađenosti između objektno orijentiranih aplikacija i SQL-a koji se temelji na relacijskom modelu (Copeland 2008)

Filozofija SQLAlchemy-ja glasi:

„SQL baze podataka se ponašaju sve manje kao kolekcije objekata što su veličina i brzina bitniji, a kolekcije objekata se ponašaju sve manje kao tablice i retci dokle god nam je potrebna apstrakcija. Cilj SQLAlchemyja je da se prilagodi ovim situacijama“ .(SQLAlchemy 2021)



Slika 10 SQLAlchemy logo

2.4. SQLite i MySQL

Kako bi objasnili i pokazali mogućnosti Jupyter notebooka koristili smo dva sistema za upravljanje bazama podataka ili DBMS (Data Base Management System). Jedan DBMS koji smo koristili je SQLite, a drugi je MySQL. Oba DBMS su relacijskog tipa, tj baziraju se na relacijskom modelu koji je trenutno najpopularniji model za DBMS Zbog svoje jednostavnosti i prilagođenosti ljudskom shvaćanju podataka i odnosa među njima. (Marić 2019)

Navedene DBMS smo odlučili koristiti zbog njihove popularnosti te zbog samih razlika koje pružaju. Glavni cilj nam je bio prikazivanje procesa integracije podataka iz baza podataka koje su nastale u tim DBMS, te premoštavanje razlika u sintaksi i strukturi podataka koristeći Pandas i SQLAlchemy

2.4.1. SQLite

SQLite je biblioteka napravita u programskom jeziku C koja se fokusira na male i brze baze podataka. SQLite je program sa otvorenim kodom što znači da se po potrebi može modoficirati za maksimalne performanse. Zbo svojih minimalističkih svojstava SQLite se koristi gdje god se radi sa strogo ograničenom memorijom. Veliki broj uređaja koirsti SQLite, te je izrazito popularan u aplikacijama za mobilne uređaje, ali i računala, zbog čega je najpopularniji DBMS na svijetu.(The SQLite Consortium, 2021)

Izvorno je objavljen 2000. godine, te je osmišljen kako bi aplikacijama omogućio prikladan način upravljanja podacima bez općih troškova koji često dolazi s namjenskim sustavima za upravljanje relacijskim bazama podataka. Popularnost SQLite se može objasniti njegovim značajkama prenosivost, jednostavnost za koirštenje, kompaktibilnost, učinkovitost i pouzdanost. (Apress Owen 2006)

SQLite je ugrađeni DBMS, što znači da umjesto da radi kao samostalni proces, on se pokreće unutar procesnog prostora aplikacije koju poslužuje, te se vrlo često kod SQLitea ne može odvojiti od koda aplikacije Na prvi pogled, takve aplikacije izgledaju kao da nemaju integriran DBMS. Jedna od prednosti poslužitelja baze podataka unutar vašeg programa je ta što nema mrežne konfiguracije i administracija. Klijent i poslužitelj se pokreću na istom procesu što olakšava administraciju baze podataka i olakšava izgradnju aplikacija. (Apress Owen 2006)

Slika 11 SQLite logo



2.4.2. MySQL (MariaDB)

MySQL je sustav za upravljanje relacijskim bazama podataka čija arhitektura se bazira na principu klijent/poslužitelj, tj MySQL služi kao poslužitelj baza podataka za proizvoljno mnogo klijenata, tj aplikacija koji traže podatke, izvršavaju upite itd. Klijenti mogu raditi na istom računalu ili se mogu spajati preko internetske veze. MySQL napisan je sa kodom otvorenog tipa, što znači da ga se po želji može mjenjati i modificirati. MySQL je također besplatan, međutim postoje i verzije koje se mogu plaćati koje nude dodatne funkcionalnosti.

(Oracle Corporation, 2021) MySQL koristi se na velikom broju platformi i operacijskih sustava te se nalazi unutar LAMP softwarea koji čine Linux, Apache, MySQL i PHP/Perl/Python

MySQL je predviđen za rad sa bazama podataka svih veličina, te je opremljen za rad sa velikim skupovima podataka. <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>

Još jedna nama bitna stvar za napomenuti je da postoji nekoliko različitih ogranaka MySQL-a, a jedan od njih je MariaDB koji ćemo kasnije koristiti u praktičnom dijelu rada. MariaDB podržava sve trenutne funkcije MySQL-a međutim također dodaje neke nove koje u ovom radu nećemo koristiti. (Oracle Corporation, 2021)

Instaliranje SQLAlchemy podrške za MariaDB

Kako bi koristili engine funkciju sqlalchemy-ja za MariaDB trebamo instalirati mysqlclient library koji je dostupan na slijedećem linku. <https://pypi.org/project/mysqlclient/> Ili jednostavno možemo pokrenuti komandu "pip install mysqlclient" u Anaconda Promptu (MariaDB, 2021)



Slika 12 MySQL logo

2.5. XAMPP

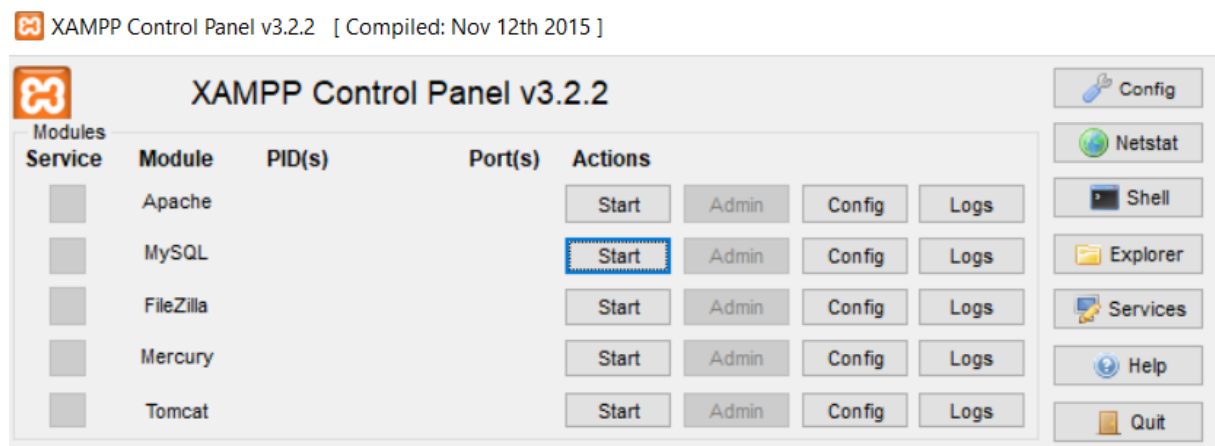
XAMPP je programski paket razvijen od strane Apache Friends organizacije, a njegova svrha je pružanje besplatnog i jednostavnog načina za instalaciju Apache servera, MariaDB DBMS i interpretera za PHP i Perl programske jezike. Navedeni programski paket pruža lagani način za instaliranje potrebne programske podrške za web poslužitelj na računalu, stoga smo ga odabrali pri izradi ovoga završnog rada. MariaDB je jedan od komponenata XAMPPA, a dolazi zajedno sa vizualnim sučeljem MyPHPAdmin. (Vmware 2021)

2.5.1. Instalacija XAMPP-a

XAMPP je izrazito jednostavno za instalirati. Kako bi obavili instalaciju, sa službene stranice <https://www.apachefriends.org/index.html> skinemo verziju aplikacije predviđenu za naš operacijski sustav, potom ga instaliramo prateći instrukcije u procesu za instaliranje. ,Vmware 2021

2.5.2. Pokretanje servera baze podataka

Za praktični rad morati ćemo imati pristup serveru baze podataka. Kako bi ostvarili pristup otvorimo XAMPP upravljačku ploču, te kliknemo na gumb „start“ desno od labela „MySQL“



Slika 13 XAMPP kontrolna ploča

3. Praktični rad

U praktičnom radu baviti ćemo se integracijom podataka u Jupyter notebooku. To ćemo obaviti tako što ćemo izraditi baze podataka u MySQL i SQLite DBMS-ovima, koristeći SQL jezik. Potom ćemo koristeći SQLAlchemy navedene baze podataka pretvoriti u podatkovne okvire (engl. data frame) kako bi sa Pandas funkcijama mogli prikazati strukturu podataka i načine za njihovu obradu i vizualizaciju. Također ćemo se koristiti matplotlib programskom bibliotekom, te njezinom funkcionalnošću prikaza grafova kako bi prikazali podatke u obliku grafova.

3.1. Rad sa MySQLom

U sljedećim primjerima pokazat ćemo princip korištenja MySQL DBMSa putem Jupyter Notebook okruženja. Sljedeće ćelije sa kodom u sebi sadrže kod za spajanje na MySQL bazu podataka, te kreiranje, uređivanje, i brisanje podataka u navedenoj bazi. Za pristup navedenoj bazi koristimo se programskim alatom XAMPP i serverom za bazu podataka koji pokrećemo lokalno.

3.1.1. Spajanje na server

U ćeliji ispod nalazi se kod za spajanje na server, koristimo funkciju iz mysql.connector programske biblioteke kako bi dali Pythonu autorizaciju da koristi server za bazu podataka. Potrebno je napomenuti da navedenu programsku biblioteku trebamo importirati sa ključnom riječi „import“. Argumenti koje koristimo pri spajanju su zadane postavke MySQL servera. Ime poslužitelja je localhost, korisnik sa kojim pristupamo tablici je „root“ i lozinka nije postavita, tako da pridružujemo vrijednost praznog polja. U slučaju da je spajanje na server baze uspješno, funkcija mysql.connector.connect vraća vrijednost „True“. Tako da smo kao malu provjeru ispod koda za spajanje napravili grananje if-else. Koje provjerava da li spomenuta funkcija daje vrijednost True ili False, te ovisno o tome ispisuje poruku „Uspješno spajanje na server :)" ili „Neuspješno spajanje na server :(“

```
In [6]: import mysql.connector
MySQLspoji = mysql.connector.connect(host="localhost", user="root", passwd="")
if (MySQLspoji):
    print("Uspješno spajanje na server :)")
else:
    print("Neuspješno spajanje na server :(")
```

Slika 14 MySQL spajanje na server baze podataka

3.1.2. Kreiranje baze podataka

U slijedećoj ćeliji koristimo se upitom za kreiranje baze podataka "skola". Za kreiranje navedene baze koristimo se SQL jezikom „CREATE DATABASE skola“. Kako bi to odradili stvaramo objekt mycursor koji sadrži vrijednosti funkcije mydb.cursor koja se koristi se za izvršavanje upita za komunikaciju s bazom podataka MySQL, te sa funkcijom execute izvršavamo SQL upit

https://www.tutorialspoint.com/python_data_access/python_mysql_cursor_object.htm

```
In [ ]: import mysql.connector
mydb = mysql.connector.connect(host="localhost", user="root", passwd="")
mycursor = mydb.cursor()
mycursor.execute("CREATE DATABASE skola")
```

Slika 15 Kreiranje baze podataka SQL upitom

3.1.3. Ispis baza podataka

Potom se koristimo upitom „SHOW DATABASES“ koji nam vraća listu svih baza podataka na serveru. Kao što možete vidjeti u slici16, u ispisu, koristeći petlju „for“ prikazane su sve baze podataka koje trenutno imamo na serveru. Među njima je baza podataka koju smo izgradili „skola“

```
In [7]: import mysql.connector
mydb = mysql.connector.connect(host="localhost", user="root", passwd="")
mycursor = mydb.cursor()
mycursor.execute("SHOW DATABASES")
for db in mycursor:
    print (db)

('harshdb',)
('information_schema',)
('iwa_2019_vz_projekt',)
('mysql',)
('performance_schema',)
('phpmyadmin',)
('skola',)
('skola_dva',)
('test',)
```

Slika 16 Ispis MySQL baza podataka

3.1.4. Kreiranje tablice

Slijedeća ćelija sadrži kod i SQL upit za kreiranje tablice "rezultati" u bazi podataka "skola". Navedena tablica sastoji se od 16 stupaca koje ćemo ukratko opisati.

Svrha navedene tablice je upis završnih rezultata učenika neke škole. Stupci koji se prikazuju su

Tablica sa atributima i tipovima podataka tablice „rezultati“

Ime stupca	Tip podatka	Tumač
Oib	BIGINT	osobni identifikacijski broj svakog učenika
Ime	Varchar	Ime učenika
Prezime	Varchar	Prezime učenika
Razred	int	redni broj razreda koji je učenik završio
Odjel	Varchar	razredni odjel kojem učenik pripada (a,b,c,d... itd)
Hj	int	Obvezni predmet: Hrvatski jezik
Mat	Int	Obvezni predmet: Matematika
Pid	Int	Obvezni predmet: Priroda i društvo
Tzk	Int	Obvezni predmet: Tjelesna kultura
Lik	Int, Null	Obvezni predmet: Likovni
Glk	Int, Null	Obvezni predmet: Glazbeni
Inf	Int, Null	Izborni predmet: Informatika
Vj	Int, Null	Izborni predmet: Vjeronauk
Eti	Int, Null	Izborni predmet: Etika
nje	Int, Null	Izborni predmet: Njemački

```

import mysql.connector
mydb = mysql.connector.connect(host="localhost", user="root", passwd="", database="skola")
mycursor = mydb.cursor()
mycursor.execute("CREATE TABLE rezultati(oib BIGINT NOT NULL,ime varchar(200), prezime varchar(200),"
                "razred int(20), odjel varchar(20), hj int(20), mat int(20), pid int(20),"
                "tzk int(20), lik int(20), glk int(20), inf int(20) NULL, vj int(20) NULL, eti int(20) NULL,"
                "nje int(20) NULL, eng int(20) NULL) CHARACTER SET 'utf8'")

```

Slika 17 Prikaz SQL upita za pravljenje tablice

3.1.5. Popunjavanje tablice

Nakon što smo kreirali tablicu trebamo unjeti podatke u nju. Za to koristimo upit „INSERT INTO rezultati“ nakon čega navodimo stupce u koje želimo unjeti vrijednosti i potom unosimo listu vrijednosti. Bitno je napomenuti da u Pythonu, ako želimo napisati izrazito dugačak string, možemo ga podijeliti na više redova ako svaki red počnemo i završimo sa znakom duplog navodnika „“. U našu tablicu unjeli smo 20 entiteta sa različitim podacima. Neki od učenika ne pohađaju neke od izbornih predmeta, u tome se slučaju u to polje uvrštava vrijednost NULL, koja se koristi kada u podatci za neki određeni atribut nisu dostupni. Ili ne postoje.

```

In [ ]: import mysql.connector
mydb = mysql.connector.connect(host="localhost", user="root", passwd="", database="skola")
mycursor = mydb.cursor()
mycursor.execute("INSERT INTO rezultati(oib, ime, prezime, razred, odjel, hj, mat, pid, tzk, lik, glk, inf, vj, eti, nje
                "(75340764935, 'Kristina', 'Tomić', 3, 'a', 5, 5, 4, 5, 5, 5, 5, 5, NULL, NULL, 5),"
                "(27439996880, 'Ana Marija', 'Jurković', 4, 'a', 5, 3, 3, 5, 5, 4, NULL, NULL, 5, 3, NULL),"
                "(81653345607, 'Luka', 'Kos', 4, 'a', 3, 4, 4, 4, 4, 5, 4, 5, NULL, 5, NULL),"
                "(73442576621, 'Kristijan', 'Babić', 4, 'a', 5, 5, 2, 5, 4, 5, NULL, NULL, 5, NULL, 4),"
                "(70999780192, 'Patrik', 'Kolarić', 3, 'a', 3, 5, 4, 4, 3, 5, 4, NULL, 5, NULL, 2),"
                "(51337695362, 'Domagoj', 'Mišić', 3, 'a', 5, 3, 2, 2, 5, 5, NULL, NULL, 4, 3, NULL),"
                "(53031118619, 'Josipa', 'Opačak', 4, 'b', 5, 4, 4, 3, 5, 5, 3, 4, NULL, 5, NULL),"
                "(20662541071, 'Karlo', 'Miletić', 4, 'b', 4, 4, 4, 4, 5, 5, 5, NULL, 5, NULL, 5),"

```

Slika 18 SQL upit za popunjavanje tablice

Ispisivanje vrijednosti tablice „rezultati“

Koristimo se SQL upitom „SELECT * FROM rezultati“ kako bi odabrali sve vrijednosti iz naše tablice „rezultati“. Navedene vrijednosti potom ispisujemo pomoću for petlje. Bitno je primjetiti kako se trenutno podatci iz tablice prikazuju u obliku polja, te izgledaju neuredno i nepregledno. Tu možemo vidjeti probleme koje predstavlja korištenje MySQL-a u objektu orijentiranom okolišu.

```

In [9]: import mysql.connector
mydb = mysql.connector.connect(host="localhost", user="root", passwd="", database="skola")
mycursor = mydb.cursor()
mycursor.execute("SELECT * FROM rezultati")
myresult = mycursor.fetchall()
for row in myresult:
    print(row)

(75340764935, 'Kristina', 'Tomić', 3, 'a', 5, 5, 4, 5, 5, 5, 5, 5, 5, None, None, 5)
(27439996880, 'Ana Marija', 'Jurković', 4, 'a', 5, 3, 3, 5, 5, 4, None, None, 5, 3, None)
(81653345607, 'Luka', 'Kos', 4, 'a', 3, 4, 4, 4, 4, 5, 4, 5, None, 5, None)
(73442576621, 'Kristijan', 'Babić', 4, 'a', 5, 5, 2, 5, 4, 5, None, None, 5, None, 4)
(70999780192, 'Patrik', 'Kolarić', 3, 'a', 3, 5, 4, 4, 3, 5, 4, None, 5, None, 2)
(51337695362, 'Domagoj', 'Mišić', 3, 'a', 5, 3, 2, 2, 5, 5, None, None, 4, 3, None)
(53031118619, 'Josipa', 'Opačak', 4, 'b', 5, 4, 4, 3, 5, 5, 3, 4, None, 5, None)
(20662541071, 'Karlo', 'Miletić', 4, 'b', 4, 4, 4, 4, 5, 5, 5, None, 5, None, 5)
(73945237386, 'Dragutin', 'Božić', 4, 'b', 3, 5, 5, 5, 5, 5, 4, None, 4, None, 2)
(67303763693, 'Iva', 'Bošnjak', 4, 'a', 5, 5, 5, 5, 5, 5, 5, 5, None, 5, None)
(25086523559, 'Bruno', 'Špoljarić', 3, 'a', 4, 3, 3, 3, 5, 4, None, None, 5, 4, None)
(34731035315, 'Burdica', 'Posavec', 4, 'a', 4, 5, 5, 4, 4, 3, 2, 4, None, None, 5)
(37985081613, 'Danijela', 'Kos', 4, 'b', 3, 3, 4, 4, 5, 5, None, 4, None, None, 4)

```

Slika 19 Kod za ispisivanje vrijednosti tablice "rezultati"

Također je bitno primjetiti da slika (gore) ne sadrži sve elemente baze, već je radi preglednost prikazan samo primjerna uzorak.

3.2. Rad sa SQLite

3.2.1. Kreiranje baze

Koristimo se programskom bibliotekom sqlite3 kako bi pokazali CRUD radnje na SQLite bazi podataka. SQLite ne zahtjeva server stoga je spajanje na bazu izrazito lako. Koristeći funkciju "connect" spajamo se na navedenu bazu, ako ona ne postoji, sqlite je kreira. Ali prije svega pridružujemo programsku biblioteku „sqlite3“ u trenutnu skriptu sa „import“

```

In [ ]: import sqlite3
conn = sqlite3.connect("bazaskola2")

```

Slika 20 SQLite spajanje na bazu "skola"

3.2.2. Stvaranje tablice

Prije nego zadamo SQL upit za pravljenje baze moramo definirati parametre za spajanje sa bazom. Prvo se spajamo na željenu bazu koristeći funkcionalnost sqlite3.connect(), a kao argument navodimo ime tablice koju želimo obraditi. U ovom slučaju to je „bazaskola2“

Zatim u objekt c uvrštavamo vrijednost funkcije conn.cursor, te nakon toga koristimo funkciju „execute“ kako bi izvršili SQL kod za kreiranje tablice pod imenom „RezultatiSkolskeGod“. U

slučaju da tablica pod istim imenom postoji, upit se ne izvršava zbog argumenta „IF NOT EXIST“ u SQL upitu.

```
In [10]: import sqlite3
conn = sqlite3.connect("bazaskola2")
c = conn.cursor()
c.execute("CREATE TABLE IF NOT EXISTS RezultatiSkolskeGod(oib integer,"
         "ime_ucenika text, prezime_ucenika text, razred integer, odjel_ucenika text,"
         " hrvatski_jezik integer, matematika integer, priroda_i_drustvo integer, likovni integer,"
         "vjeronauk integer, glazbeni integer, engleski integer)")
conn.commit()
conn.close()
```

Slika 21 SQL upit za kreiranje tablice

Kako bi predočili sadržaj tablice objasniti ćemo značenje pojedinog stupca

Ova tablica prikazuje atribute i tipove podataka tablice „RezultatSkolskeGod“

Ime stupca	Tip podatka	Tumač
Oib	INT	osobni identifikacijski broj svakog učenika
Ime_ucenika	Text	Ime učenika
Prezime_ucenika	Text	Prezime učenika
Razred	Int	redni broj razreda koji je učenik završio
Odjel_ucenika	Varchar	razredni odjel kojem učenik pripada (a,b,c,d... itd)
Hrvatski_jezik	Int	Obvezni predmet: Hrvatski jezik
Matematika	Int	Obvezni predmet: Matematika
Priroda i društvo	Int	Obvezni predmet: Priroda i društvo
Vjeronauk	Int	Obvezni predmet: Vjeronauk
Likovni	Int	Obvezni predmet: Likovni
Engleski	Int	Obvezni predmet: Engleski
Glazbeni	Int	Obvezni predmet: Glazbeni

Bitno je primjetiti da u ovoj tablici ne postoje NULL vrijednosti jer su svi predmeti označeni kao Obvezni, također je bitno primjetiti da ako usporedimo tablicu „RezultatiSkolskeGod“ i „rezultati“ da baza podataka „rezultati“ ima više stupaca. Navednu razliku smo namjerno napravili kako bi prikazali mogućnosti spajanja neujednačenih tablica sa Pandas programskim alatom.

3.2.3. Popunjavanje tablice i ispisivanje vrijednosti

Koristeći slijedeći SQL kod popunjavamo tablicu sa vrijednostima koje smo zadali.

```
In [ ]: import sqlite3
conn = sqlite3.connect("bazaskola2")
c = conn.cursor()
c.execute("INSERT INTO RezultatiSkolskeGod VALUES (82246012449, 'Mario', 'Matić', 3, 'a', 2, 3, 4, 2, 2, 5, 3),")
"(87359162978, 'Ivan', 'Tokić', 4, 'a', 5, 3, 4, 3, 3, 2, 5),"
"(84708087224, 'Jan', 'Špoljar', 3, 'a', 5, 5, 2, 5, 4, 4, 5),"
"(30018474546, 'Kristina', 'Car', 3, 'b', 5, 4, 5, 5, 5, 5, 5),"
"(94058913622, 'Tin', 'Korak', 4, 'a', 4, 5, 4, 5, 5, 5, 4),"
"(21146138396, 'Tomislav', 'Radić', 3, 'b', 4, 2, 5, 5, 4, 5, 5),"
"(89378436711, 'Željka', 'Dalić', 4, 'b', 3, 3, 4, 5, 4, 5, 2),"
"(37096685803, 'Josipa', 'Novak', 4, 'a', 5, 5, 5, 5, 5, 5, 5),"
"(85014710648, 'Petar', 'Mullig', 4, 'b', 4, 4, 4, 5, 5, 5, 4),"
"(56579276955, 'Stjepan', 'Rebić', 3, 'b', 5, 5, 2, 5, 5, 4, 2),"
"(94249385377, 'Zlatko', 'Janković', 3, 'a', 2, 2, 2, 4, 4, 4, 5)"
conn.commit()
conn.close()
```

Potom koristeći se SQL upitom „SELECT * FROM RezultatiSkolskeGod“ odabiremo sve vrijednosti u tablici te ih sa funkcijom Print ispisujemo vrijednost koju vraća funkcija „fetchall“

```
In [11]: import sqlite3
conn = sqlite3.connect("bazaskola2")
c = conn.cursor()
c.execute("SELECT * FROM RezultatiSkolskeGod ")
print(c.fetchall())
conn.commit()
conn.close()

[(82246012449, 'Mario', 'Matić', 3, 'a', 2, 3, 4, 2, 2, 5, 3), (87359162978, 'Ivan', 'Tokić', 4, 'a', 5, 3, 4, 3, 3, 2, 5), (84708087224, 'Jan', 'Špoljar', 3, 'a', 5, 5, 2, 5, 4, 4, 5), (30018474546, 'Kristina', 'Car', 3, 'b', 5, 4, 5, 5, 5, 5, 5), (94058913622, 'Tin', 'Korak', 4, 'a', 4, 5, 4, 5, 5, 5, 4), (21146138396, 'Tomislav', 'Radić', 3, 'b', 4, 2, 5, 5, 4, 5, 5), (89378436711, 'Željka', 'Dalić', 4, 'b', 3, 3, 4, 5, 4, 5, 2), (37096685803, 'Josipa', 'Novak', 4, 'a', 5, 5, 5, 5, 5, 5, 5), (85014710648, 'Petar', 'Mullig', 4, 'b', 4, 4, 4, 5, 5, 5, 4), (56579276955, 'Stjepan', 'Rebić', 3, 'b', 5, 5, 2, 5, 5, 4, 2), (94249385377, 'Zlatko', 'Janković', 3, 'a', 2, 2, 2, 4, 4, 4, 5)]
```

3.3. SQLAlchemy i Pandas

U ćeliji ispod koristit ćemo sqlalchemy za pretvaranje baze podataka u podatkovni okvir (dataframe), te Pandas za čitanje i prikazivanje podataka. Argument koji dajemo funkciji "sqlalchemy.create_engine" se može objasniti na slijedeći način: mysql+mysqldb://ime_usera_baze:password_usera@ime_poslužitelja:port/ime_baze?en kriptacija

3.3.1. Čitanje tablica - pandas

Nakon toga koristimo Pandas funkciju `read_sql_table`, te kao argumente navodimo ime tablice "rezultati" i varijablu "engine" te je pozivamo, i kao rezultat toga dobijamo ispis "rezultat".

```
In [30]: import pandas as pd
import sqlalchemy
engine = sqlalchemy.create_engine('mysql+mysqldb://root:@localhost:3306/skola?charset=utf8mb4')
mysqldataframe = pd.read_sql_table('rezultati',engine)
mysqldataframe
```

```
Out[30]:
```

	oib	ime	prezime	razred	odjel	hj	mat	pid	tzk	lik	glk	inf	vj	eti	nje	eng
0	75340764935	Kristina	Tomić	3	a	5	5	4	5	5	5	5.0	5.0	NaN	NaN	5.0
1	27439996880	Ana Marija	Jurković	4	a	5	3	3	5	5	4	NaN	NaN	5.0	3.0	NaN
2	81653345607	Luka	Kos	4	a	3	4	4	4	4	5	4.0	5.0	NaN	5.0	NaN
3	73442576621	Kristijan	Babić	4	a	5	5	2	5	4	5	NaN	NaN	5.0	NaN	4.0
4	70999780192	Patrik	Kolarić	3	a	3	5	4	4	3	5	4.0	NaN	5.0	NaN	2.0
5	51337695362	Domagoj	Mišić	3	a	5	3	2	2	5	5	NaN	NaN	4.0	3.0	NaN
6	53031118619	Josipa	Opačak	4	b	5	4	4	3	5	5	3.0	4.0	NaN	5.0	NaN
7	20662541071	Karlo	Miletić	4	b	4	4	4	4	5	5	5.0	NaN	5.0	NaN	5.0
8	73945237386	Dragutin	Božić	4	b	3	5	5	5	5	5	4.0	NaN	4.0	NaN	2.0
9	67303763693	Iva	Bošnjak	4	a	5	5	5	5	5	5	5.0	5.0	NaN	5.0	NaN
10	25086523559	Bruno	Špoljarić	3	a	4	3	3	3	5	4	NaN	NaN	5.0	4.0	NaN
11	34731035315	Đurđica	Posavec	4	a	4	5	5	4	4	3	2.0	4.0	NaN	NaN	5.0
12	37985081613	Danijela	Kos	4	b	3	3	4	4	5	5	NaN	4.0	NaN	NaN	4.0

Slika 22 Korištenje Pandas za ispis tablice

Kao što možemo vidjeti na slici gore, Ispis tablice sa Pandas programskim alatom prikazuje preglednu tablicu koja je estetski puno ugodnija oku nego ispis u polja kao što smo imali na prošlim primjerima.

3.3.2. Pretvaranje SQLite baze podataka u podatkovni okvir

Isti postupak ponavljamo u slijedećoj ćeliji. Samo u njoj koristimo bazu Sqlite podataka, te imajući to na umu trebamo promijeniti argument engine-a. Za više detalja o postavka engine-a pogledajte slijedeći link: <https://docs.sqlalchemy.org/en/14/core/engines.html#sqlite>

```
In [13]: enginesqlite = sqlalchemy.create_engine('sqlite:///bazaskola2')
dfsqliite = pd.read_sql_table('RezultatiSkolskeGod',enginesqlite)
dfsqliite
```

```
Out[13]:
```

	oib	ime_ucenika	prezime_ucenika	razred	odjel_ucenika	hrvatski_jezik	matematika	priroda_i_drustvo	likovni	vjeronauk	glazbeni	engleski
0	82246012449	Mario	Matić	3	a	2	3	4	2	2	5	3
1	87359162978	Ivan	Tokić	4	a	5	3	4	3	3	2	5
2	84708087224	Jan	Špoljar	3	a	5	5	2	5	4	4	5
3	30018474546	Kristina	Car	3	b	5	4	5	5	5	5	5
4	94058913622	Tin	Korak	4	a	4	5	4	5	5	5	4
5	21146138396	Tomislav	Radić	3	b	4	2	5	5	4	5	5
6	89378436711	Željka	Dalić	4	b	3	3	4	5	4	5	2
7	37096885803	Josipa	Novak	4	a	5	5	5	5	5	5	5
8	85014710648	Petar	Mulig	4	b	4	4	4	5	5	5	4
9	56579276955	Stjepan	Rebić	3	b	5	5	2	5	5	4	2
10	94249385377	Zlatko	Janković	3	a	2	2	2	4	4	4	5

3.3.3.Prikaz određenih polja podatkovnog okvira

U slijedećem polju koristimo SQL upit kako bi odabrali vrijednosti polja za sve učenike iz odjela „a“. Koristimo SQL upit "SELECT * FROM RezultatiSkolskeGod WHERE odjel_ucenika = 'a'"

Te nakon toga koristimo pandas funkciju „read_sql_query“ kojoj dajemo argumente:

- Upit koji želimo da se izvrši
- Podatkovni okvir nad kojim radimo upit

```
In [29]: upit = "SELECT * FROM RezultatiSkolskeGod WHERE odjel_ucenika = 'a'"
podatkovniokvir = pd.read_sql_query(upit, enginesqlite)
podatkovniokvir
```

```
Out[29]:
```

	oib	ime_ucenika	prezime_ucenika	razred	odjel_ucenika	hrvatski_jezik	matematika	priroda_i_drustvo	likovni	vjeronauk	glazbeni	engleski
0	82246012449	Mario	Matić	3	a	2	3	4	2	2	5	3
1	87359162978	Ivan	Tokić	4	a	5	3	4	3	3	2	5
2	84708087224	Jan	Špoljar	3	a	5	5	2	5	4	4	5
3	94058913622	Tin	Korak	4	a	4	5	4	5	5	5	4
4	37096885803	Josipa	Novak	4	a	5	5	5	5	5	5	5
5	94249385377	Zlatko	Janković	3	a	2	2	2	4	4	4	5

Tablica sa svim učenicima iz odjela „a“ se ispisuje u čitljivom pandas formatu.

3.3.4.Mjenjanje imena stupaca podatkovnog okvira

Nad naša dva podatkovna okvira želimo odraditi operaciju konkatencije. Pandas odrađuje navedenu operaciju tako da doda vrijednosti jedne tablice drugoj, imajući u obzir imena stupaca. Kako bi ispravno odradili operaciju konkatencije, moramo preimenovati stupce sa istim podacima, tako da su isti u obje tablice. Npr stupac „ime_ucenika“ ćemo preimenovati u „ime“ kako i odgovaralo drugoj tablici.

Preimenovanje podataka obavljamo sa funkcijom .columns, koja prima tipove podatka „string“ za imena stupaca koje ćemo preimenovati.


```
In [25]: sqlitedataframe = pd.read_sql_table('RezultatiSkolskeGod', enginesqlite)
sqlitedataframe.columns = ['oib', 'ime', 'prezime', 'razred', 'odjel', 'hj', 'mat', 'pid', 'lik', 'vj', 'glk', 'eng']
sqlitedataframe
```

```
Out[25]:
```

	oib	ime	prezime	razred	odjel	hj	mat	pid	lik	vj	glk	eng
0	82246012449	Mario	Matić	3	a	2	3	4	2	2	5	3
1	87359162978	Ivan	Tokić	4	a	5	3	4	3	3	2	5
2	84708087224	Jan	Špoljar	3	a	5	5	2	5	4	4	5
3	30018474546	Kristina	Car	3	b	5	4	5	5	5	5	5
4	94058913622	Tin	Korak	4	a	4	5	4	5	5	5	4
5	21146138396	Tomislav	Radić	3	b	4	2	5	5	4	5	5
6	89378436711	Željka	Dalić	4	b	3	3	4	5	4	5	2
7	37096685803	Josipa	Novak	4	a	5	5	5	5	5	5	5
8	85014710648	Petar	Mulig	4	b	4	4	4	5	5	5	4
9	56579276955	Stjepan	Rebić	3	b	5	5	2	5	5	4	2
10	94249385377	Zlatko	Janković	3	a	2	2	2	4	4	4	5

Slika 23 Konkatencija tablica 1

3.3.5. Spajanje tablica – konkatencija

Koristimo se Pandas funkcijom "concat" kako bi spojili retke iz oba podatkovna okvira u jedan podatkovni okvir. Primjetimo da se u mysql podatkovnom okviru nalaze neki stupci koji nisu prisutni u sqlite podatkovnom okviru. To su stupci "inf, eti, nje". Navedeni stupci predstavljaju izborne predmete koje učenici mogu, a i ne moraju upisati, dok oni u drugoj školi nisu dostupni. Stoga dodani entiteti u atributima "inf, eti, nje" imaju vrijednost NaN.

```
In [31]: konkatencija = pd.concat([mysqldataframe, sqlitedataframe])
konkatencija
```

```
Out[31]:
```

	oib	ime	prezime	razred	odjel	hj	mat	pid	tzk	lik	glk	inf	vj	eti	nje	eng
0	75340764935	Kristina	Tomić	3	a	5	5	4	5.0	5	5	5.0	5.0	NaN	NaN	5.0
1	27439996880	Ana Marija	Jurković	4	a	5	3	3	5.0	5	4	NaN	NaN	5.0	3.0	NaN
2	81653345607	Luka	Kos	4	a	3	4	4	4.0	4	5	4.0	5.0	NaN	5.0	NaN
3	73442576621	Kristijan	Babić	4	a	5	5	2	5.0	4	5	NaN	NaN	5.0	NaN	4.0
4	70999780192	Patrik	Kolarić	3	a	3	5	4	4.0	3	5	4.0	NaN	5.0	NaN	2.0
5	51337695362	Domagoj	Mišić	3	a	5	3	2	2.0	5	5	NaN	NaN	4.0	3.0	NaN
6	53031118619	Josipa	Opačak	4	b	5	4	4	3.0	5	5	3.0	4.0	NaN	5.0	NaN

Slika 24 Konkatencija tablica 2

Na slici gore ne vidi se dobro prikaz konkatencije zbog veličine tablice sa učenicima, ali na slici dolje se može vidjeti prijelaz iz jednog podatkovnog okvira u drugi ako gledamo u redni broj entiteta

26	43058300799	Sara	Sokač	3	a	3	4	4	4.0	5	5	5.0	5.0	NaN	5.0	NaN
27	10724354009	Jura	Jurić	3	b	5	5	5	4.0	5	5	5.0	NaN	5.0	NaN	5.0
28	63440264838	Martina	Devčić	3	b	5	5	5	5.0	5	5	4.0	NaN	5.0	NaN	4.0
29	24565860200	Lidija	Marjanović	4	a	4	5	5	4.0	5	5	3.0	5.0	NaN	5.0	NaN
0	82246012449	Mario	Matić	3	a	2	3	4	NaN	2	5	NaN	2.0	NaN	NaN	3.0
1	87359162978	Ivan	Tokić	4	a	5	3	4	NaN	3	2	NaN	3.0	NaN	NaN	5.0
2	84708087224	Jan	Špoljar	3	a	5	5	2	NaN	5	4	NaN	4.0	NaN	NaN	5.0
3	30018474546	Kristina	Car	3	b	5	4	5	NaN	5	5	NaN	5.0	NaN	NaN	5.0
4	94058913622	Tin	Korak	4	a	4	5	4	NaN	5	5	NaN	5.0	NaN	NaN	4.0
5	21146138396	Tomislav	Radić	3	b	4	2	5	NaN	5	5	NaN	4.0	NaN	NaN	5.0
6	89378436711	Željka	Dalić	4	b	3	3	4	NaN	5	5	NaN	4.0	NaN	NaN	2.0
7	37096685803	Josipa	Novak	4	a	5	5	5	NaN	5	5	NaN	5.0	NaN	NaN	5.0
8	85014710648	Petar	Mulig	4	b	4	4	4	NaN	5	5	NaN	5.0	NaN	NaN	4.0
9	56579276955	Stjepan	Rebić	3	b	5	5	2	NaN	5	4	NaN	5.0	NaN	NaN	2.0
10	94249385377	Zlatko	Janković	3	a	2	2	2	NaN	4	4	NaN	4.0	NaN	NaN	5.0

3.3.6. Računanje prosjeka po predmetu (stupcu) – mean

Koristeći podatkovni okvir koji smo dobili sa konkatencijom, možemo izračunati prosječne ocijene na pojedinom predmetu, za to koristimo funkciju „mean“

```
In [32]: meanhj = konkatencija["hj"].mean()
meaninf = konkatencija["inf"].mean()
print("Prosjek zaključenih ocijena iz hrvatskog jezika je " , meanhj , " dok je iz informatike " , meaninf)
Prosjek zaključenih ocijena iz hrvatskog jezika je 4.121951219512195 dok je iz informatike 4.090909090909091
```

Slika 25 Računanje prosjeka po predmetu (stupcu) – mean

3.3.7. Računanje prosjeka učenika (retci)

U slijedećoj tablici na podatkovni okvir "konkatencija" dodajemo stupac "prosjek". Prosjek je vrijednost koju vraća funkcija "mean", a kako bi pravilno izračunali prosjek trebali smo specificirati koji stupci ulaze u računicu. Također kako bi izračunali prosjek vrijednosti stupca, koristimo argument za funkciju mean "axis=1"

```
In [44]: konkatenacija["prosjek"] = konkatenacija[['hj', 'mat', 'pid', 'tzk', 'lik', 'glk', 'inf', 'vj', 'eti', 'nje', 'eng']].mean(axis=1)
konkatenacija
```

```
Out [44]:
```

	oib	ime	prezime	razred	odjel	hj	mat	pid	tzk	lik	glk	inf	vj	eti	nje	eng	prosjek
0	75340764935	Kristina	Tomić	3	a	5	5	4	5.0	5	5	5.0	5.0	NaN	NaN	5.0	4.888889
1	27439996880	Ana Marija	Jurković	4	a	5	3	3	5.0	5	4	NaN	NaN	5.0	3.0	NaN	4.125000
2	81653345607	Luka	Kos	4	a	3	4	4	4.0	4	5	4.0	5.0	NaN	5.0	NaN	4.222222
3	73442576621	Kristijan	Babić	4	a	5	5	2	5.0	4	5	NaN	NaN	5.0	NaN	4.0	4.375000
4	70999780192	Patrik	Kolarić	3	a	3	5	4	4.0	3	5	4.0	NaN	5.0	NaN	2.0	3.888889
5	51337695362	Dornagoj	Mišić	3	a	5	3	2	2.0	5	5	NaN	NaN	4.0	3.0	NaN	3.625000
6	53031118619	Josipa	Opačak	4	b	5	4	4	3.0	5	5	3.0	4.0	NaN	5.0	NaN	4.222222
7	20662541071	Karlo	Miletić	4	b	4	4	4	4.0	5	5	5.0	NaN	5.0	NaN	5.0	4.555556

3.3.8. Prosjek ocjena učenika - prosjek svih učenika

U sljedećoj ćeliji ćemo napisati kod za izračun prosjeka svih vrijednosti atributa "prosjek", te ćemo poredati entitete učenika po prosjeku. Za sortiranje učenika koirstimo funkciju „sort.values“. Nakon toga ćemo prikazati 3 učenika sa najvišim i 3 sa najnižim prosjekom. Za prikaz prva (x) polja tablice koristimo se funkcijom „head“, a za prikaz zdanjih (x) koristimo se funkcijom tail(). Gdje je x broj prvih/zadnjih stupaca koje želimo da se ispišu.

```
In [72]: ukupniprosjek = konkatenacija["prosjek"].mean()
print ("Srednja vrijednost posjeka svih učenika je: ", ukupniprosjek)
poredani_ukupniprosjek = konkatenacija.sort_values("prosjek", ascending=False)
najveci_prosjek = poredani_ukupniprosjek.head(3)
najveci_prosjek[["prezime", "ime", "prosjek"]]
```

Srednja vrijednost posjeka svih učenika je: 4.2537262872628725

```
Out [72]:
```

	prezime	ime	prosjek
7	Novak	Josipa	5.000000
9	Bošnjak	Iva	5.000000
0	Tomić	Kristina	4.888889

```
In [73]: najmanji_prosjek = poredani_ukupniprosjek.tail(3)
najmanji_prosjek[["prezime", "ime", "prosjek"]]
```

```
Out [73]:
```

	prezime	ime	prosjek
10	Janković	Zlatko	3.285714
0	Matić	Mario	3.000000
15	Bašić	Paula	3.000000

3.4. Matplotlib

Matplotlib je jedna od programskih biblioteka za prezentaciju podataka te ćemo pomoću nje prikazati horizontalni stupčasti graf prosjeka ocijena svih učenika. Crvena ocrtkava linija će prikazivati srednji prosjek svih učenika. Matplotlib programsku biblioteku pozivamo sa „import“

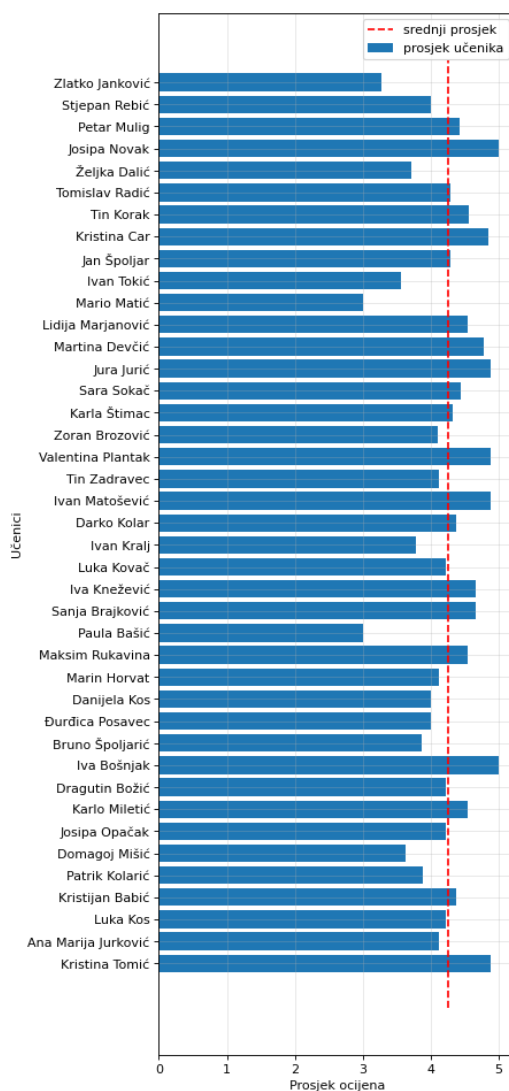
```
In [139]: import matplotlib.pyplot as plt
import numpy as np
from matplotlib.pyplot import figure
konkatenacija["ime_prezime"] = konkatenacija["ime"] + " " + konkatenacija["prezime"]
%matplotlib inline
imeprezime = konkatenacija["ime_prezime"]
prosijek = konkatenacija["prosijek"]
y_os = np.arange(len(imeprezime))
figure(figsize=(5, 15), dpi=80)
plt.yticks(y_os, imeprezime)
plt.ylabel("Učenici")
plt.xlabel("Prosijek ocijena")
plt.grid(alpha=0.3)
plt.vlines(ukupniprosijek, -2, 41, linestyle='--', color='red', label="srednji prosijek")
plt.barh(y_os, prosijek, label="prosijek učenika")
plt.legend()
```

Kako bi prikazali ime i prezime zajedno pored prosjeka ocijena, stvaramo novi stupac i nazvamo ga „ime_prezime“ te u njega spajamo vrijednost polja „ime“, razmak „ “ i vrijednost polja „prezime“.

Nakon toga prikupljamo ostale potrebne podatke, kao što je npr, rednji prosjek, te koristimo matplotlib funkcije kako bi odredili parametre grafa. Npr:

- `Plt.yticks` – pozicionira ime i prezime pored odgovarajućeg stupca
- `Plt.ylabel`, `plt.xlabel` – imenovanje y i x osi
- `Plt.grid` – generiranje „mreže“ na grafu
- `Plt.vlines` – vertikalna linija, u našem će slučaju biti linija koja prikazuje srednji prosijek
- `Plt.barh` – horizontalni stupci, u našem slučaju će prikazivati prosjek ocjena učenika
- `Plt.legend` – legenda grafa

Nakon što smo sve definirali, generira se graf koji možemo vidjeti na slici 26



Slika 26 Graf prosjeka ocjena izrađen u Matplotlibu

4. Kritički prikaz

Smatram da je ovaj završni rad bio iznimno zanimljiv, Jupyter Notebook ima jako veliki potencijal kod prikazivanja strukture podataka i učenja samog Pythona jer može pokretati male djelove koda. Ako bih opet radio sličan rad koristio bih izvore podataka koje se mogu naći na internetu jer su mnogo zanimljiviji te bih proveo više vremena istražujući razne programske biblioteke.

5. Zaključak

Jupyter Notebook je sjajan alat za prikaz strukture podataka i pored svoje mogućnosti da kod pokreće dio po dio također se može koristiti sa bezbroj programskih biblioteka koje

uvelike proširuju te mogućnosti. Glavni programski alati koje možemo koristiti za podatkovnu znanost u Jupyter Notebooku i Pythonu općenito, su SQLAlchemy i Pandas koji nude ogromne mogućnosti kod čitanja, obrađivanja i kreiranja zapisa u baze podataka

Popis literature

- Driscoll Mike, bez.dat, Jupyter Notebook Introduction (pristupito 19.9.2021)
<https://realpython.com/jupyter-notebook-introduction/>
- Jupyter Team,2015, Jupyter Notebook, Jupyter Notebook introduction ,(Pristupito:19.9.2021) <https://jupyter-notebook.readthedocs.io/en/stable/notebook.html>
- Shirokov Slava, 2015, GitHub + Jupyter Notebooks = <3,(Pristupito:19.9.2021)
<https://github.blog/2015-05-07-github-jupyter-notebooks-3/>
- Debian.org, N.D. ,(Pristupito:19.9.2021)
<https://www.debian.org/doc/manuals/aptitude/pr01s02.en.html>
- Andaconda.org, 2021 ,Anaconda Navigator, ,(Pristupito:19.9.2021)
<https://docs.anaconda.com/anaconda/navigator/>
- Anaconda.org 2021, ,(Pristupito:19.9.2021) Anconda individual edition
<https://docs.anaconda.com/anaconda/>
- The pandas development team, 2008-2021,Installation, ,(Pristupito:19.9.2021)
https://pandas.pydata.org/docs/getting_started/install.html
- The pandas development team, 2008-2021,Getting started, ,(Pristupito:19.9.2021)
https://pandas.pydata.org/pandas-docs/stable/getting_started/overview.html
- The pandas development team, 2008-2021,About, ,(Pristupito:19.9.2021)
<https://pandas.pydata.org/about/index.html>
- The pandas development team, 2008-2021,Intro tutorials, ,(Pristupito:19.9.2021)
https://pandas.pydata.org/docs/getting_started/intro_tutorials/01_table_oriented.html
- Copeland Rick, 2008, Essential SQLAlchemy , O'Reilly media
SQLAlchemy 2021, About ,(Pristupito:19.9.2021) , <https://www.sqlalchemy.org/>
- Marić, Matej 2019, Usporedba relacijske baze podatka i ne relacijske baze podatka
- The SQLite Consortium, 2021, What Is SQLite?,(Pristupito:19.9.2021)
<https://www.sqlite.org/index.html>
- Owens Michael, 2006, The Definitive Guide to SQLite
- Oracle Corporation, 2021, What is MySQL?,(Pristupito:19.9.2021),

<https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>

MariaDB, 2021, MariaDB versus MySQL: Compatibility, (Pristupito:19.9.2021)

<https://mariadb.com/kb/en/mariadb-vs-mysql-compatibility/>

Vmware 2021, About, , (Pristupito:19.9.2021),
<https://www.apachefriends.org/about.html>

TutorialPoint,2021, Python MySQL – Cursor Object, (Pristupito:19.9.2021)

https://www.tutorialspoint.com/python_data_access/python_mysql_cursor_object.htm

Popis slika

Slika 1 Instaliranje Jupyter Notebook-a sa PIP-om	3
Slika 2 Instalacija Anaconda	5
Slika 3 Anaconda Navigator.....	5
Slika 4 Tipovi ćelija u Jupyter Notebook-u	6
Slika 5 Jupyter browser	6
Slika 6 Izgled Jupyter Notebooka	7
Slika 7 Izgled Jupyter Notebook Markdown ćelija	8
Slika 8 Jupyter Notebook sa slikom unutar ćelije	9
Slika 9 Pandas podržava mnogo različitih vrsta podataka	11
Slika 10 SQLAlchemy logo	12
Slika 11 SQLite logo	13
Slika 12 MySQL logo	14
Slika 13 XAMPP kontrolna ploča	15
Slika 14 MySQL spajanje na server baze podataka	16
Slika 15 Kreiranje baze podataka SQL upitom	17
Slika 16 Ispis MySQL baza podataka	17
Slika 17 Prikaz SQL upita za pravljenje tablice	19
Slika 18 SQL upit za popunjavanje tablice	19
Slika 19 Kod za ispisivanje vrijednosti tablice "rezultati"	20
Slika 20 SQLite spajanje na bazu "skola"	20
Slika 21 SQL upit za kreiranje tablice.....	21
Slika 22 Korištenje Pandas za ispis tablice	23
Slika 23 Konkatencija tablica 1	25
Slika 24 Konkatencija tablica 2	25
Slika 25 Računanje prosjeka po predmetu (stupcu) – mean	26
Slika 26 Graf prosjeka ocjena izrađen u Matplotlibu.....	29

Slika 10 SQLAlchemy logo <http://www.sqlalchemy.org>

Slika 9 Pandas podržava mnogo različitih vrsta podataka -

https://pandas.pydata.org/docs/getting_started/intro_tutorials/02_read_write.html

Slika.12 MySQL logo - <https://en.wikipedia.org/w/index.php?curid=67634535>

SQLite logo - <https://commons.wikimedia.org/w/index.php?curid=11675072>