

Lumen - Data Science 2023 Tehnička Dokumentacija

Sadržaj

Sadržaj	2
1. Uvod	3
2. Organizacija	4
3. Aplikacija	7

1. Uvod

Glazba je univerzalni jezik koji postoji tisućama godina, a stvara se korištenjem širokog spektra instrumenata. Razumijevanje različitih vrsta instrumenata i njihovih jedinstvenih karakteristika bitno je za svakoga tko je zainteresiran za proučavanje i izvođenje glazbe.

Ova dokumentacija pruža tehnički pregled sustava kojeg smo koristili i aplikacije.

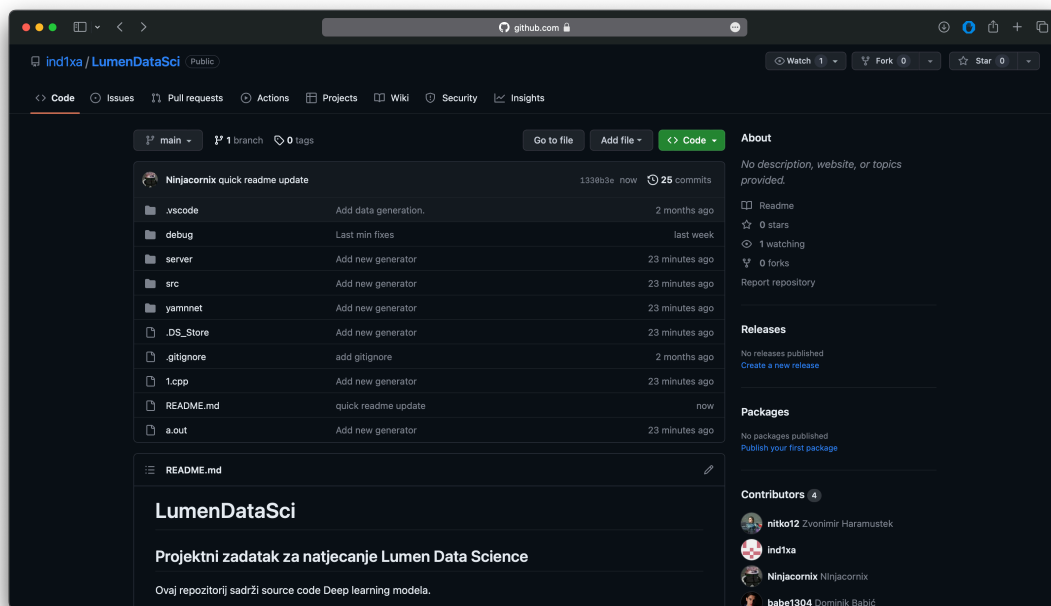
2. Organizacija

Ovo poglavlje posvećujemo internoj organizaciji projekta i radu u grupi.

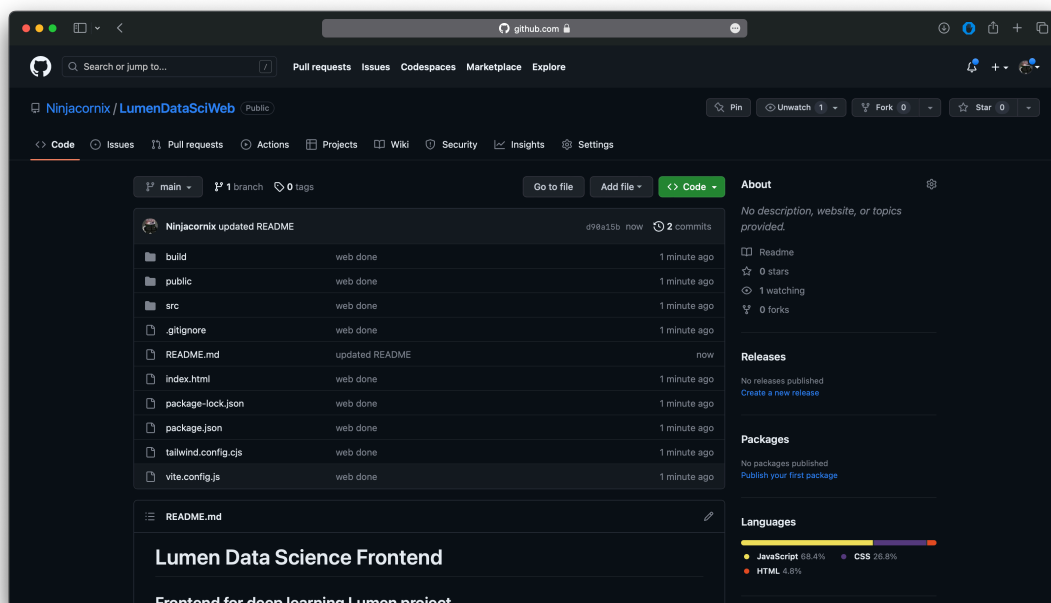
1. Organizacija koda
2. Organizacija podataka
3. Organiziranje treniranja

2.1 ORGANIZACIJA KODA

Sav kod smo objavljivali na Github repozitoriju "<https://github.com/ind1xa/LumenDataSci>". Tu se nalazi deep learning source code, server, a frontend na "<https://github.com/Ninjacornix/LumenDataSciWeb>".



Slika 2.1 GitHub repozitorij

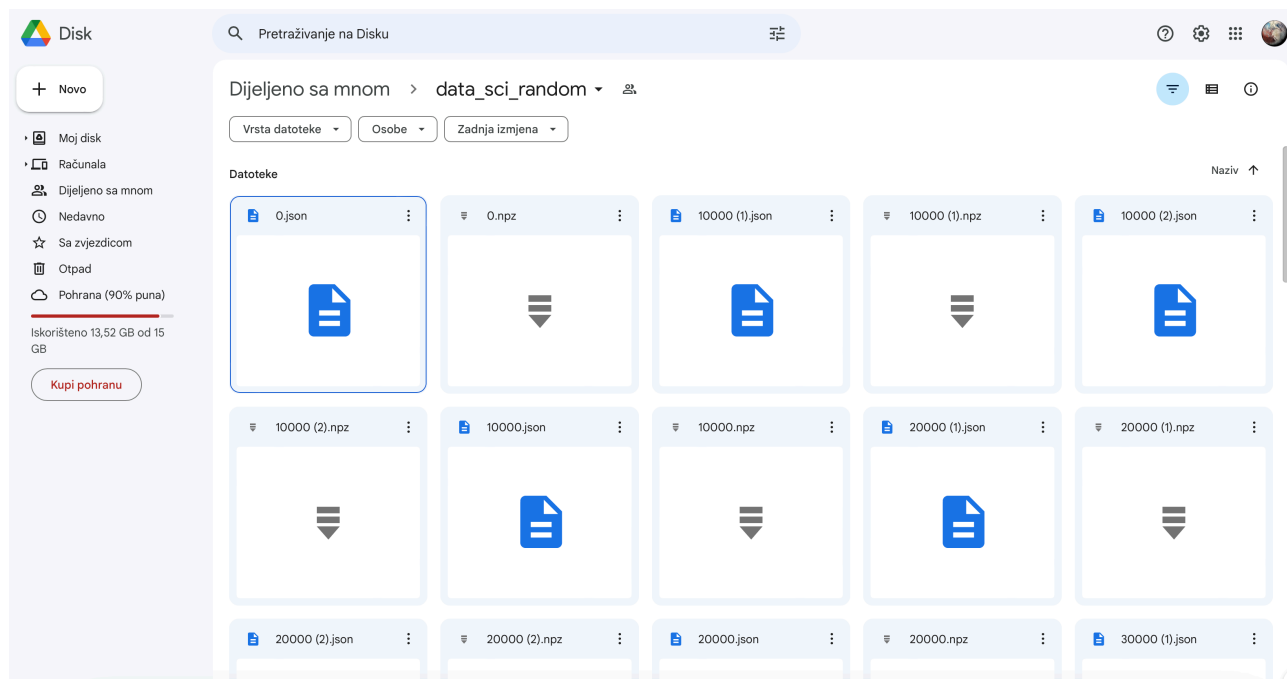


Slika 2.2 Frontend GitHub repozitorij

2.2 ORGANIZACIJA PODATAKA

Podatke smo prebacili na Google Drive jer nam je bila potrebna velika količina prostora za augmentaciju audiozapisa. Da bi osigurali dovoljno mjesta, odlučili smo kupiti dodatni prostor na Google Driveu. Nakon što smo završili s augmentacijom, prebacili smo audiozapise na računala koje smo iznajmili, opremljena sa nekoliko grafičkih kartica, kako bismo mogli brzo i učinkovito obraditi podatke.

Zbog količine podataka, bilo nam je potrebno oko **200GB** prostora na računalu. Dane podatke smo prebacili u RAM memoriju zbog bržeg treniranja. Dio podataka smo čuvali i na Microsoftovom OneDriveu.



Slika 2.3 Dijeljeni Google Drive sa augmentiranim podacima

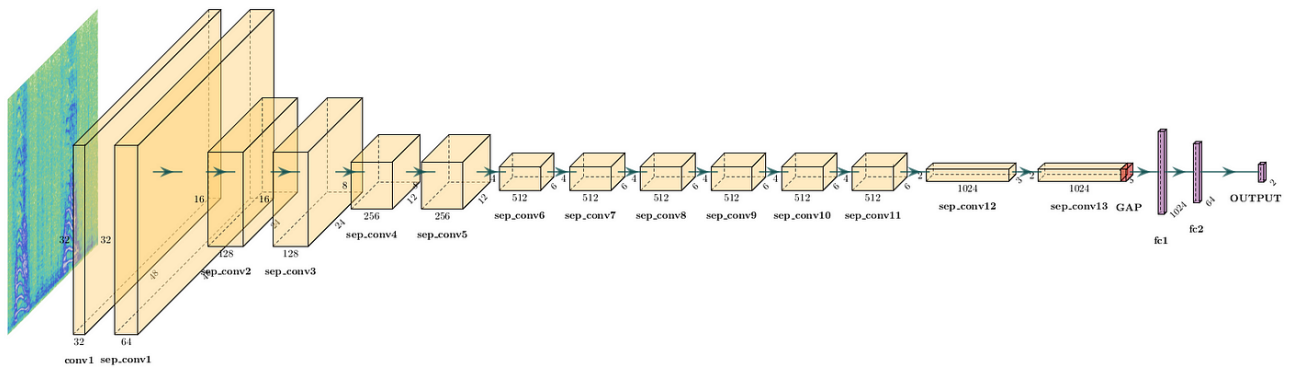
2.3 ORGANIZACIJA TRENIRANJA

Prilikom treniranja koristili smo fiksnu podjelu **90/10**. Što znači da je 90% skupova podataka otišlo na treniranje, a 10% za validaciju. Prilikom treniranja koristili smo “čiste” podatke i augmentirane tako da dobijemo veću robusnost. Podaci su na kraju testirani koristeći dani validation dataset.



Slika 2.4 Podjela podataka

Za treniranje koristili smo Keras. Keras služi kao interface za TensorFlow library koji pojednostavljuje proces treniranja i validacije modela. Budući da smo imali dosta pokušaja i različitih eksperimenata, na kraju smo se odlučili za YAMNet. YAMNet je predtrenirana neuralna mreža koju smo dodatno trenirali na zvukovima danima kao podaci za treniranje.

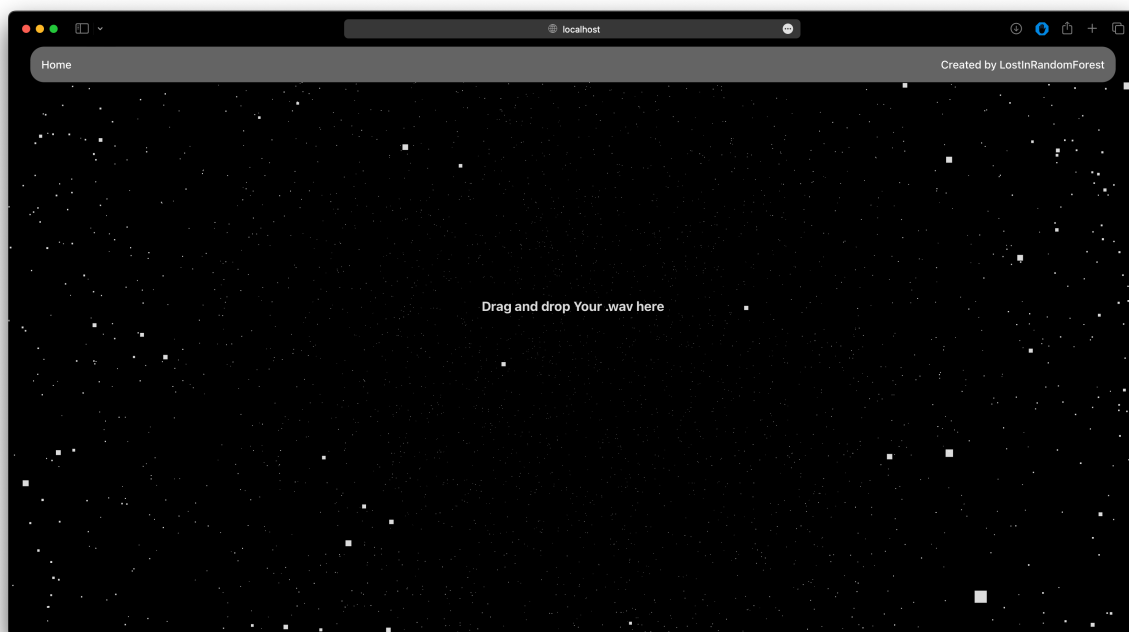


Slika 2.5 YAMNet arhitektura

3. Aplikacija

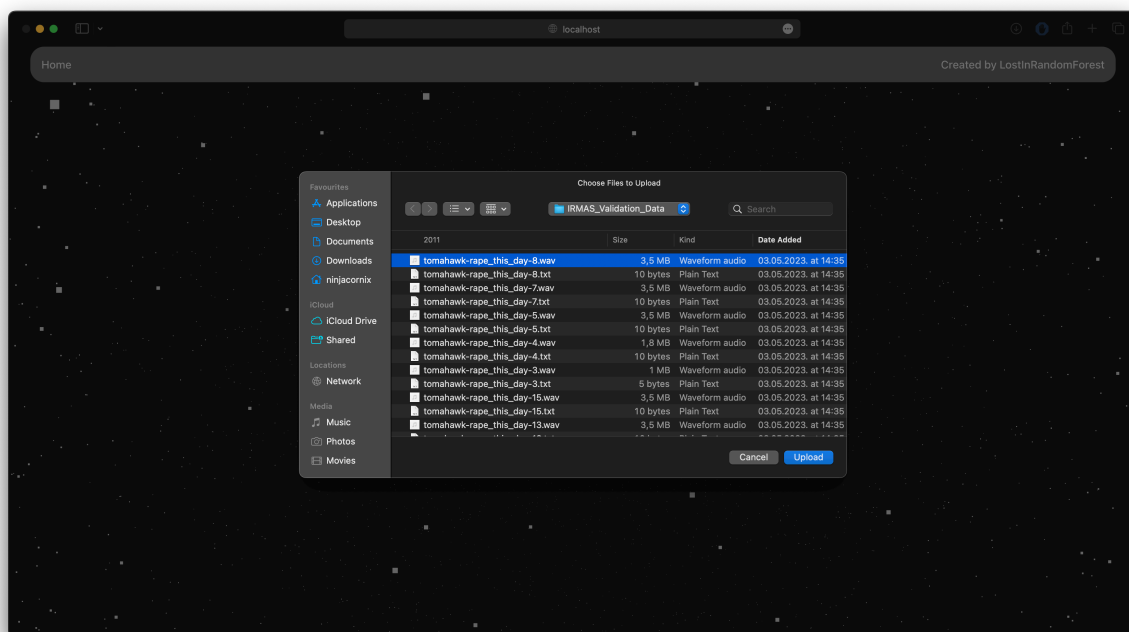
U ovom poglavlju ćemo govoriti o funkcionalnosti naše aplikacije te proći tehničke detalje. Glavna ideja je da aplikacija bude jednostavna korisniku za koristiti. Nije potrebno lokalno imati podatke već jednostavno otići na stranicu “lostinrandomforest.click” i uploadati audiozapis po želji.

FRONTEND



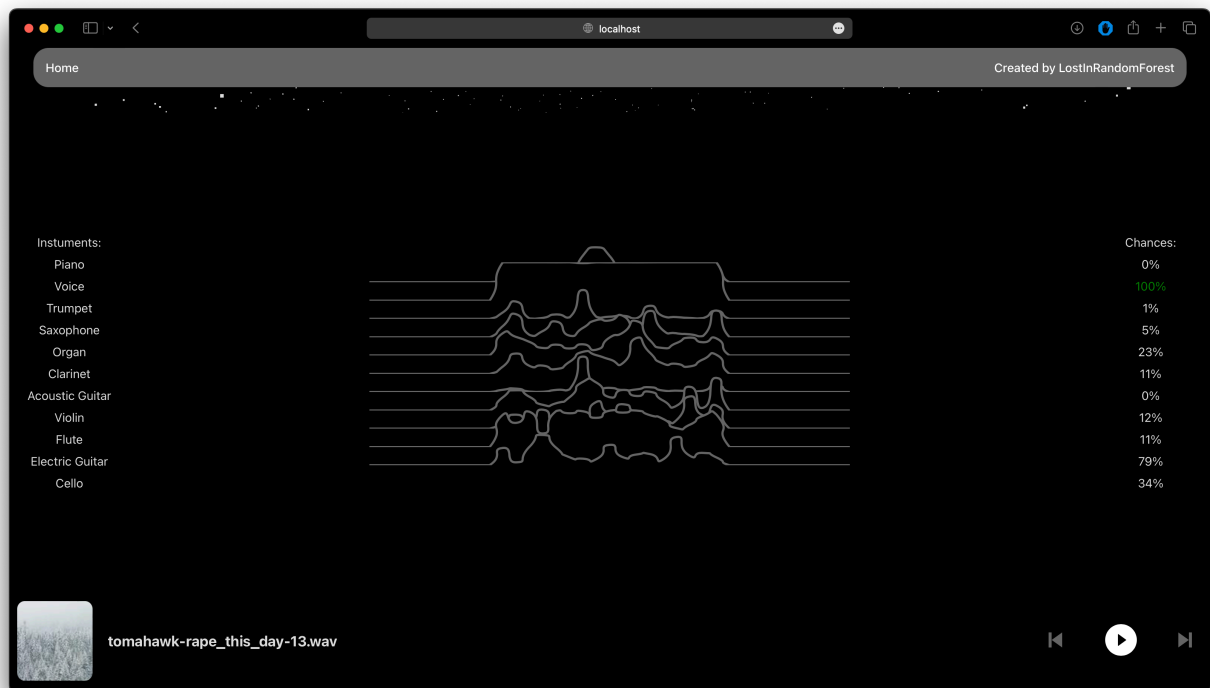
Slika 3.1 Izgled aplikacije nakon otvaranja

Pri ulasku na web stranicu korisnik dobije ovaj pogled. Dokumente koje želimo analizirati treba povući i ispustiti na tekst “Drag and drop Your .wav here” ili odabrati klikom na taj tekst.



Slika 3.2 Odabir audiozapisa za obradu

Nakon što odaberemo datoteku, ona se prenosi na server gdje započinje njena obrada.



Slika 3.3 Prikaz segmenata u kojem su dektektirani određeni instrumenti

Nakon uspješne obrade vidimo na kojim segmentima se pojavljuje određeni instrument kao i ukupnu šansu da se taj pojavljuje.

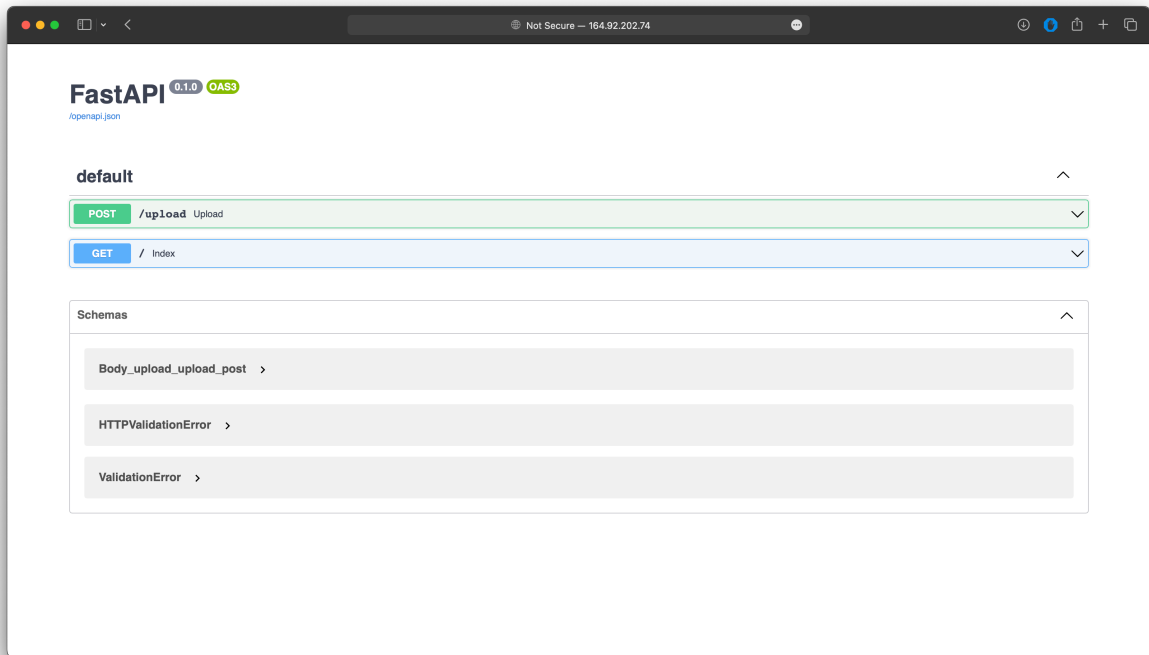
Šanse obojane zelenom bojom odgovaraju Hammingovom rezultatu koji je konkretno u našem slučaju izračunat na **85.05**.

Na dnu stranice imamo audioplayer uz čiju pomoć možemo poslušati dokument koji smo dali na obradu.

API I BACKEND

Za komunikaciju između aplikacije i backenda koristili smo rest API. Navedeni API razvili smo koristeći Pythonov FastAPI koji je vrlo jednostavan za korištenje.

Jednom kad je API podignut postoji endpoint na koji se aplikacija spaja i šalje POST request na endpoint: /upload. Taj endpoint koristi se da se na njega pošalje formData koji sadrži naš file.



Slika 3.4 Prikaz API endpointa