



Srednja šola za strojništvo,
mehatroniko in medije

Raziskovalna naloga

GENERIRANJE PESMI S POMOČJO UMETNE INTELIGENCE

Avtor:

Patrik GOBEC, M-4. c

Mentor:

mag. Matej VEBER, univ. dipl. inž.

Celje, januar 2022

IZJAVA*

Mentor Matej Veber v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom Generiranje glasbe s pomočjo umetne inteligence, katere avtor je Patrik Gobec:

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 15.4.2022

žig šole

Podpis mentorja

Podpis odgovorne osebe

*

POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.

POVZETEK

V tej raziskovalni nalogi obravnavam umetno inteligenco in njeno uporabo. Poudarek je na uporabi umetne inteligence na glasbenem področju, na katerem sem preučil in preizkusil že obstoječe primere za generiranje glasbe. Naučil sem se tudi, kako s knjižnico Magenta ustvariti in trenirati nove modele ter z njimi generirati nove pesmi. Nalogo sem zaključil z anketo, ki daje vpogled v mnenje dijakov o umetni inteligenci, in s kvizom, ki je pokazal stopnjo razločevanja med napisano in generirano glasbo.

Ključne besede: umetna inteligenca, programiranje, Python, Anaconda, Magenta, glasba

ABSTRACT

The research paper deals with artificial intelligence and its applications. It focuses on the use of AI in the field of music, where we studied and tested the existing AIs for generating music. We have also learned how to use Magenta to create and train new models and how to generate new songs with them. The paper concludes with a survey, which was conducted in order to gain insight into students' opinions on AI, and a quiz to check whether students could tell the difference between the written and generated music.

Keywords: artificial intelligence, programming, Python, Anaconda, Magenta, music

KAZALO

1 UVOD	9
1.1 OPIS PROBLEMATIKE	9
1.2 HIPOTEZE	9
1.3 RAZISKOVALNE METODE	9
1.4 CILJI	10
2 UMETNA INTELIGENCA OZIROMA AI	11
2.1 KAJ JE AI?	11
2.2 PODROČJA UPORABE UMETNE INTELIGENCE	11
2.2.1 Strojno učenje (angl. <i>machine learning</i>)	12
2.2.2 Globoko učenje (angl. <i>deep learning</i>)	13
2.2.3 Podatkovna znanost (angl. <i>data science</i>)	14
2.2.4 Robotika (angl. <i>robotics</i>)	14
2.2.5 Nevronska mreža (angl. <i>neural network</i>)	16
2.3 UPORABA AI	17
2.3.1 Uporaba umetne inteligence dandanes	17
2.3.2 Možne uporabe umetne inteligence	19
3 UMETNA INTELIGENCA V GLASBI	20
3.1 JUKEBOX	20
3.1.1 Stiskanje zvoka	21
3.1.2 Generiranje zvoka	22
3.1.3 Nabor podatkov za učenje	23
3.1.4 Besedilo	23
3.1.5 Avtor in žanr	24
3.1.6 Omejitve	25
3.2 MUSENET	26

3.2.1 Omejitve MuseNeta	26
3.2.2 Delovanje MuseNeta.....	26
4 GENERIRANJE GLASBE S POMOČJO MAGENTE.....	28
4.1 IDEJA.....	28
4.2 ODLOČANJE	28
4.3 PRIPRAVA DELOVNEGA OKOLJA.....	28
4.4 PRIPRAVA PODATKOV ZA UČENJE.....	34
4.5 TRENIRANJE MODELA IN GENERIRANJE GLASBE	38
4.6 UGOTOVITVE	43
4.6.1 Izbiranje podatkov za učenje	43
4.6.2 Izbiranje umetne inteligence	43
5 MNENJE DIJAKOV O UMETNI INTELIGENCI	44
5.1 ANKETA.....	44
5.2 KVIZ.....	51
6 HIPOTEZE.....	53
7 ZAKLJUČEK.....	54
8 SEZNAM UPORABLJENIH VIROV	55
9 PRILOGE	57

KAZALO SLIK

Slika 1: Področja uporabe umetne inteligence [11]	11
Slika 2: Delitev strojnega učenja [5].....	12
Slika 3: Primer globokega učenja [12].....	13
Slika 4: Humanoidni robot [1]	15
Slika 5: Zgradba nevronske mreže [4]	16
Slika 6: Primeri uporabe umetne inteligence v vsakdanjem življenju [2]	17
Slika 7: Stiskanje zvoka [10]	21
Slika 8: Generiranje zvoka [6]	22
Slika 9: Razvrščanje izvajalcev po žanrih [9].....	24
Slika 10: Pythonova spletna stran	28
Slika 11: Namestitev programa Python	29
Slika 12: Spletna stran orodja Anaconda	30
Slika 13: Anaconda (namestitev).....	30
Slika 14: Ustvarjanje novega programskega okolja.....	31
Slika 15: Izbira parametrov za novo okolje	31
Slika 16: Odpiranje terminala znotraj novega okolja	32
Slika 17: Ukaz za namestitev knjižnice Magenta	32
Slika 18: Izpis terminala po uspešni namestitvi Magente.....	33
Slika 19: Izgled delovne mape po dodajanju knjižnice	34
Slika 20: Izgled delovne mape	35
Slika 21: Izgled delovne mape	35
Slika 22: Odpiranje terminala iz mape.....	36
Slika 23: Aktivacija programskega okolja	36
Slika 24: Ukaz za pretvorbo datotek MIDI v »note sequence« datoteko	37
Slika 25: Vse možnosti umetne inteligence znotraj knjižnice Magenta	38
Slika 26: Odpiranje datoteke README.md.....	39
Slika 27: Datoteka README.md.....	40
Slika 28: Ukaz v terminalu znotraj delovne mape in programskega okolja	40
Slika 29: Ukaz za treniranje modela	41
Slika 30: Ukaz in izpis terminala za nadzorovanje treniranja.....	41
Slika 31: URL v brskalniku	41
Slika 32: Ukaz za generiranje pesmi.....	42

Slika 33: Kviz	51
----------------------	----

KAZALO GRAFIKONOV

Grafikon 1: Ali ste že slišali za umetno inteligenco?	44
Grafikon 2: Ali mislite, da se umetna inteligenca hitro razvija?	45
Grafikon 3: Ali menite, da bo umetna inteligenca kdaj pametnejša od nas?	45
Grafikon 4: Ali menite, da bomo zaradi umetne inteligence imeli probleme z zaposlovanjem?	46
Grafikon 5: Menite, da bi se lahko ali se morda že srečujemo z umetno inteligenco?	47
Grafikon 6: Ali mislite, da bi lahko umetno inteligenco uporabili za pisanje glasbe?	48
Grafikon 7: Ali mislite, da bi lahko umetna inteligenca ustvarjala boljšo glasbo od nas?	48
Grafikon 8: Kakšne prednosti ima umetna inteligenca, če jih seveda ima?	49
Grafikon 9: Ali mislite, da že obstaja umetna inteligenca za ustvarjanje glasbe?	49
Grafikon 10: Ali bi poslušali glasbo, ki jo je napisala umetna inteligenca?	50
Grafikon 11: Rezultati kviza	52

KRATICE IN SIMBOLI Z RAZLAGO

- AI – *artificial intelligence* (sl. umetna inteligenca).
- MIDI – Musical Instrument Digital Interface (tj. tehnično standardizirani komunikacijski protokol, ki povezuje široko paleto elektronskih glasbil, računalnikov in sorodnih zvočnih naprav, za predvajanje, urejanje in snemanje glasbe).
- ZIP – format arhivske datoteke, ki podpira stiskanje podatkov brez izgub.

1 UVOD

Z umetno inteligenco se iz dneva v dan pogosteje srečujemo. Uporabljamo jo neprestano pri vsakodnevnih opravilih, kot je brskanje po spletu, pri uporabi virtualnih asistentov in spletnem nakupovanju. Mnogi menijo, da umetna inteligenca ne bo nikoli sposobna narisati slike, kot jo je Leonardo da Vinci, ali skladati simfonije, kot jo je ustvaril npr. Ludwig van Beethoven. A časi se spreminjajo.

1.1 OPIS PROBLEMATIKE

Naloga, ki sem si jo zastavil, je povezana s področjem glasbe, ki je že precej izčrpano, saj je zelo težko napisati nekaj novega, kar bi pritegnilo ljudi in hkrati bilo dovolj različno, da bi za svoje delo imel avtorske pravice. Tako sem se odločil, da pomanjkanje idej za pisanje pesmi rešim s pomočjo umetne inteligence, ki bi nam lahko hitro predlagala kaj novega. Sprašujem se, ali lahko s pomočjo umetne inteligence in spletnih platform ustvarimo nove glasbene vzorce, ki bodo zanimivi za poslušalce.

1.2 HIPOTEZE

Za doseg zastavljenega cilja sem postavil naslednje hipoteze:

- HIPOTEZA 1: Umetna inteligenca lahko ustvarja glasbo.
- HIPOTEZA 2: Spletne strani za generiranje glasbe obstajajo.
- HIPOTEZA 3: Ustvarjanje umetne inteligence za generiranje glasbe je preprosto.
- HIPOTEZA 4: Možno je uporabiti predustvarjene modele za generiranje glasbe.
- HIPOTEZA 5: Umetna inteligenca ne more generirati petja.

1.3 RAZISKOVALNE METODE

V svoji raziskovalni nalogi sem uporabil metodi raziskovanja, in sicer:

- anketiranje, pri čemer sem dijake povprašal o njihovem znanju in mnenju o umetni inteligenci ter njeni uporabi v glasbi;
- raziskovanje, med katerim sem po spletu iskal in analiziral podatke o umetni inteligenci, našel že obstoječe primere umetne inteligence za ustvarjanje glasbe in ustvaril svoj model.

1.4 CILJI

Cilj moje raziskovalne naloge je ustvariti melodijo, ki bi jo napisala umetna inteligenca in bi jo lahko izboljšali s svojim znanjem glasbe.

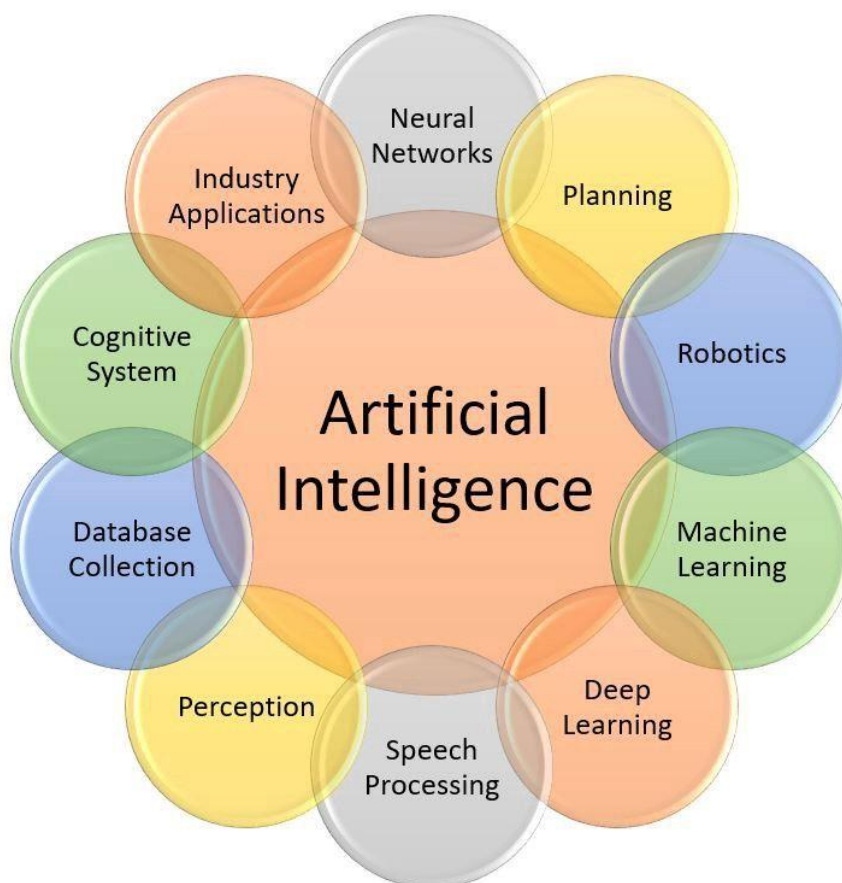
2 UMETNA INTELIGENCA OZIROMA AI

2.1 KAJ JE AI?

AI (angl. *artificial intelligence*) ali umetna inteligenca je interdisciplinarna veda, ki proučuje naprave, ki so sposobne posnemati človekovo razmišljanje, sklepanje, planiranje, učenje, znanje, komuniciranje in percepcijo. Področje AI-ja se stalno spreminja [3] [14].

2.2 PODROČJA UPORABE UMETNE INTELIGENCE

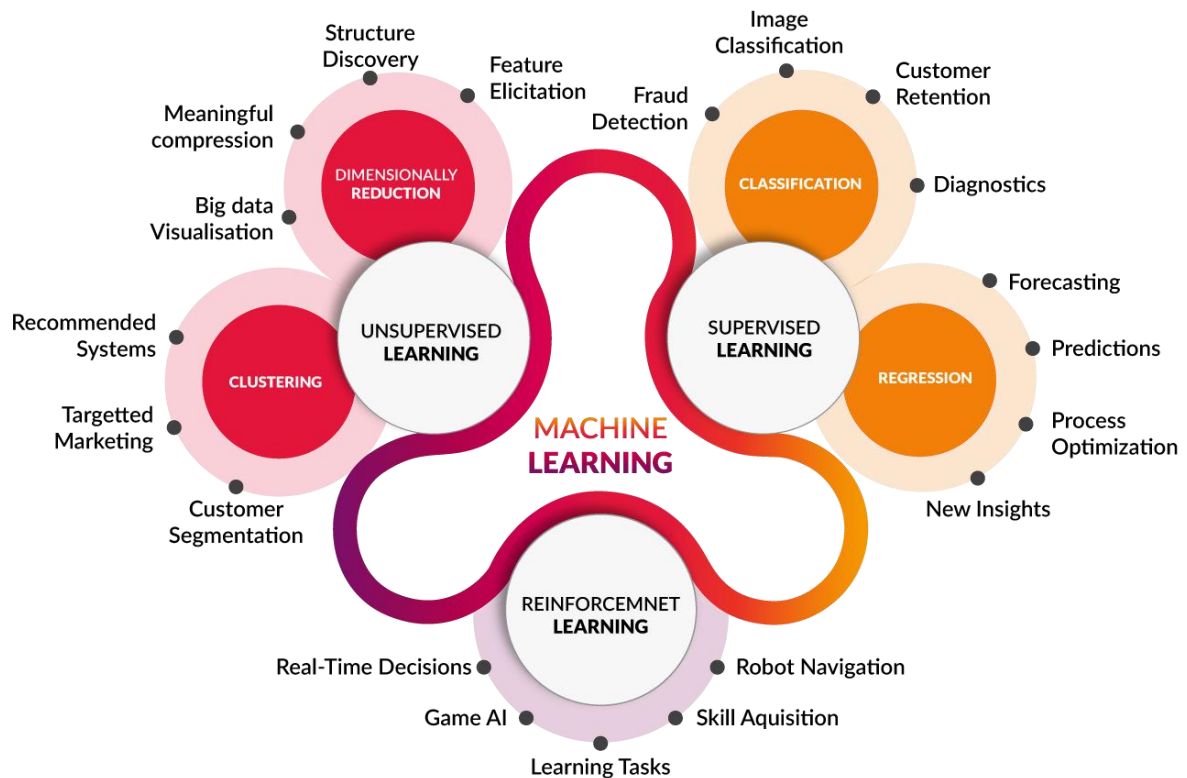
Pomembna področja uporabe umetne inteligence so strojno učenje (angl. *machine learning*), globoko učenje (angl. *deep learning*), podatkovna znanost (angl. *data science*), robotika (angl. *robotics*) in nevronske mreže (angl. *neural networks*).



Slika 1: Področja uporabe umetne inteligence [11]

2.2.1 Strojno učenje (angl. *machine learning*)

Strojno učenje pomeni pridobivanje znanja na podlagi izkušenj. Pri tem sistemu ne gre za učenje na pamet, ampak za iskanje pravil znotraj pridobljenih podatkov. Tako se z večanjem količine podatkov tudi izboljšuje uspešnost sistema. Predstavniki tega sistema so nevronske mreže, metoda podpornih vektorjev in skriti model Markova [20].



Slika 2: Delitev strojnega učenja [5]

Strojno učenje delimo na nadzorovano učenje (angl. *supervised learning*), nenadzorovano učenje (angl. *unsupervised learning*) in vzpodbujevalno učenje (angl. *reinforcement learning*).

Nadzorovano učenje (angl. *supervised learning*) – algoritem uči stroj s podanimi pari vhodnih in želenih podatkov. Pri tem želene izhode določi učitelj oz. človek, ki nadzira učenje.

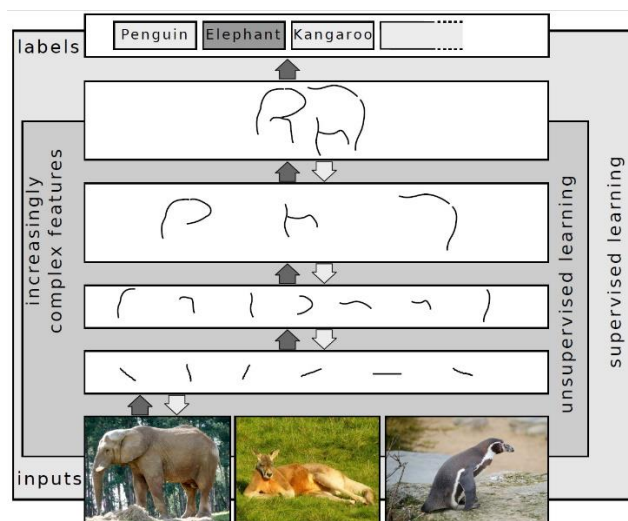
Nenadzorovano učenje (angl. *unsupervised learning*) – algoritem razdeli dane vhodne podatke po svojih kriterijih v kategorije, ki imajo lastne značilnosti. To se imenuje rojenje (angl.

clustering). Število kategorij in njihove značilnosti izlušči algoritem iz vhodnih podatkov brez nadzora učitelja.

Vzpodbujevalno učenje (angl. *reinforcement learning*) – algoritem uči z nagrajevanjem in kaznovanjem.

2.2.2 Globoko učenje (angl. *deep learning*)

Globoko učenje je del širše družine algoritmov strojnega učenja, ki temeljijo na umetnih nevronske omrežjih. Globoko učenje se uporablja v več plasteh za postopno ekstrakcijo funkcij višje ravni iz neobdelanega vhoda (npr. prepoznavo govora in slik, razumevanje besedila, pisanje besedila in glasbe, algoritem za igranje namiznih iger itd.). Npr. pri obdelavi slik (Slika 3) lahko nižje plasti prepoznajo robove, medtem ko lahko višje plasti identificirajo koncepte, pomembne za človeka, kot so številke, črke ali obrazi [13] [16].



Slika 3: Primer globokega učenja [12]

Pridevnik »globok« iz globokega učenja se nanaša na kompleksnost matematičnega modela oz. število slojev nevronske mreže. Večja zmogljivost sodobnih računalnikov je raziskovalcem omogočila povečanje kompleksnosti algoritma, s čimer so lahko dosegli ravni, ki so kvantitativno in tudi kvalitativno drugačne kot prej [13] [16].

Eden odmevnejših dosežkov globokega učenja je osvojitve naslova svetovnega prvaka v igri go, in sicer leta 2016. Gre za strateško miselno igro, pri kateri je možnih nešteto različnih potez, zato pomnjenje vseh potez iz preteklih partij ni mogoče. Globoka nevronska mreža AlphaGo se je morala znajti v novih, še nevidenih okoliščinah. AlphaGo so sprva seznanili s preteklimi

mojstrskimi partijami igre go, nato pa so jo pustili, da se je sama izpopolnjevala. Tako je po nekaj mesecih premagala aktualnega svetovnega prvaka Leeja Sedola z rezultatom tri proti nič [17].

2.2.3 Podatkovna znanost (angl. *data science*)

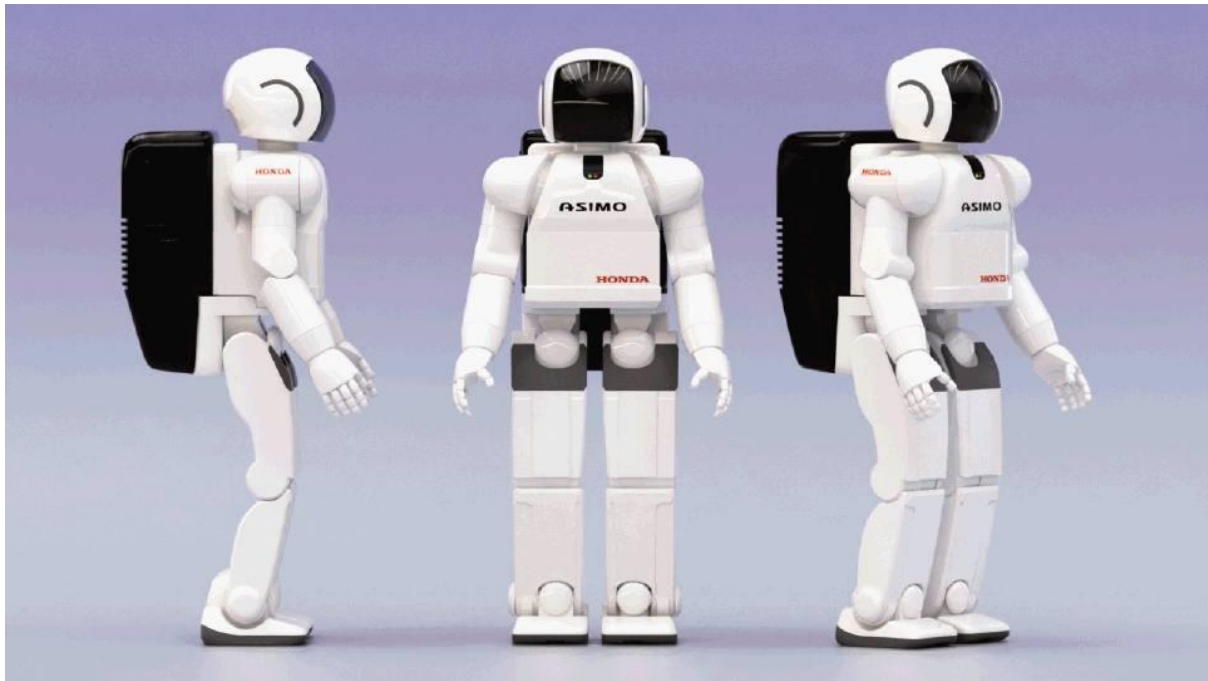
Podatkovna znanost je nov krovni izraz, ki zajema strojno učenje in statistiko ter nekatere vidike računalništva, vključno z algoritmi, shranjevanjem podatkov in razvojem spletnih aplikacij [13].

Koncept podatkovne znanosti je poenotenje statistike, analize podatkov, informatike in drugih z njimi povezanih metod, da bi lahko razumeli in analizirali dejanske pojave s podatki. Podatkovna znanost uporablja tehnike in teorije, ki izhajajo s številnih področij v okviru matematike, statistike, računalništva, informacijske znanosti in strokovnega znanja. Podatkovni znanstvenik ustvarja programsko kodo in jo združuje s statističnim znanjem, da ustvari vpogled v podatke [15].

2.2.4 Robotika (angl. *robotics*)

Robotika je interdisciplinarna veja računalništva in inženiringa, ki se ukvarja z načrtovanjem, konstrukcijo, delovanjem in uporabo robotov. Cilj robotike je oblikovati stroje, ki lahko pomagajo ljudem. Robotika združuje področja strojništva, elektrotehnike, mehatronike, elektronike, matematike itd.

Robotika razvija stroje, ki lahko nadomestijo ljudi in posnemajo človeška dejanja. Roboti se uporabljajo v številnih situacijah in za različne namene. Najpogosteje se uporabljajo v nevarnih okoljih (za pregledovanje radioaktivnih materialov, odkrivanje in deaktiviranje bomb itd.), proizvodnih procesih, v katerih ljudje ne morejo preživeti (npr. v vesolju, pod vodo ipd.). Roboti so po videzu lahko različni, nekateri nas spominjajo na ljudi (humanoidni roboti). Pravi ti (Slika 4) lahko imitirajo hojo, dviganje, govor ali katero koli drugo človeško dejavnost.



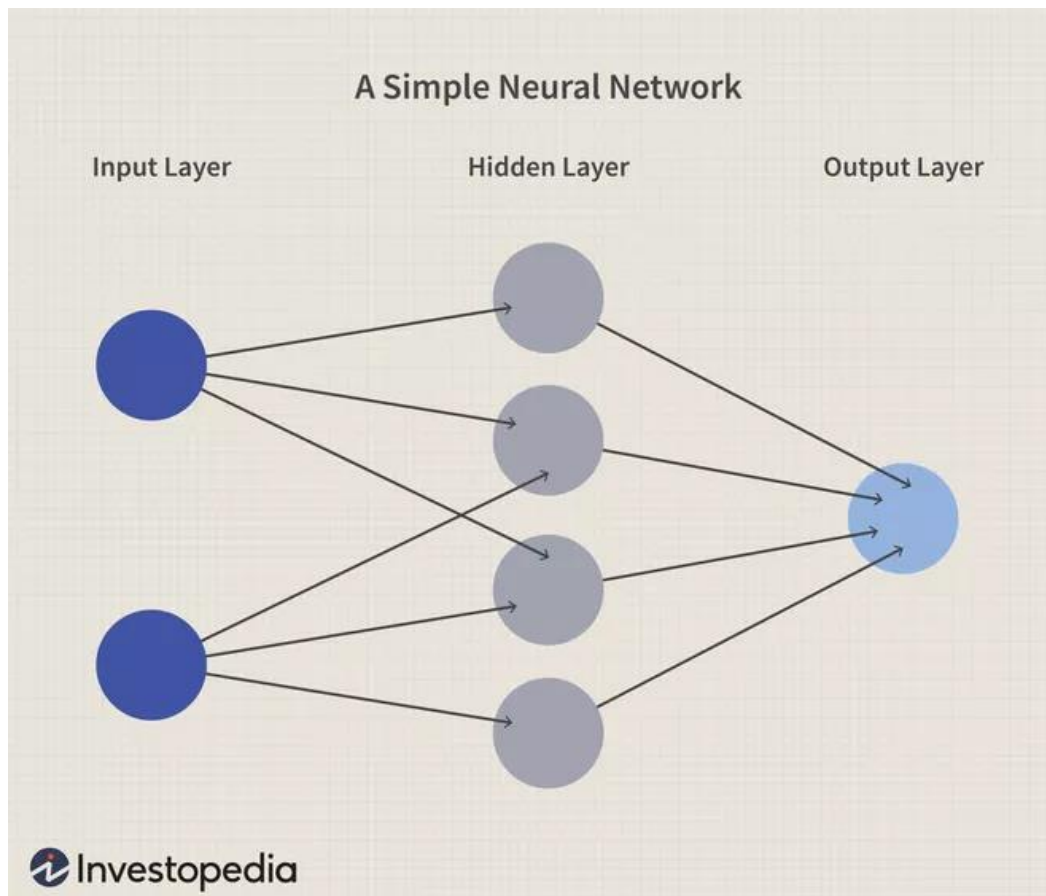
Slika 4: Humanoidni robot [1]

Nekateri roboti za delovanje potrebujejo uporabniški vnos, medtem ko drugi delujejo samostojno. Uporaba robotov, ki lahko delujejo avtonomno oz. samostojno, se je začela šele v 20. stoletju. Skozi zgodovino so različni učenjaki, izumitelji, inženirji in tehniki pogosto domnevali, da bodo roboti nekega dne lahko posnemali človeško vedenje in opravljali naloge na človeku podoben način.

Danes je robotika hitro rastoče področje, saj se tehnološki napredek nadaljuje. Raziskovanje, načrtovanje in izgradnja novih robotov pa služijo različnim, bodisi domačim, komercialnim ali vojaškim namenom. Številni roboti so zgrajeni za opravljanje nalog, ki so nevarne ljudem, kot npr. deaktiviranje bomb, iskanje preživelih v nestabilnih ruševinah ter raziskovanje rudnikov in ladijskih razbitin. Robotika se uporablja tudi v znanosti, tehnologiji, inženiringu in matematiki kot učni pripomoček [19].

2.2.5 Nevronska mreža (angl. *neural network*)

Nevronska mreža je niz algoritmov in posnema delovanje človeških možganov. Sposobna je prepoznati odnose iz niza podatkov. Nevronska omrežja se tako nanašajo na sisteme nevronov, ki so lahko organske ali umetne narave. Nevronska omrežja se neprestano prilagajajo vходу, tako da lahko omrežje ustvari najboljši možni rezultat, ne da bi bilo treba ponovno zasnovati izhodna merila. Koncept tovrstnih omrežij izhaja iz umetne inteligence [18].



Slika 5: Zgradba nevronske mreže [4]

2.3 UPORABA AI

2.3.1 Uporaba umetne inteligence dandanes



Slika 6: Primeri uporabe umetne inteligence v vsakdanjem življenju [2]

Prepoznavamo naslednja področja uporabe umetne inteligence:

- Spletno nakupovanje in oglaševanje: Umetna inteligenca se uporablja za priporočila, prilagojena posamezniku, pri tem analizira njegova pretekla iskanja in nakupe ter drugo vedenje na spletu. Umetna inteligenca je izjemnega pomena za trgovanje, saj omogoča optimiziranje izdelkov, načrtovanje zalog, boljšo logistiko itd.
- Spletno iskanje: Iskalniki se učijo iz ogromnih količin podatkov, ki jih vnašajo uporabniki, da zagotavljajo relevantne rezultate iskanja.

- Digitalni osebni asistenti: Pametni telefoni uporabljajo umetno inteligenco, da lahko zagotovijo posamezniku čim bolj prilagojen izdelek. Virtualni asistenti, ki odgovarjajo na vprašanja, nudijo predloge na različnih področjih in pomagajo organizirati dan, so danes nekaj običajnega.
- Strojni prevodi: Programska oprema za prevajanje, ki obdeluje pisno ali govorjeno besedilo, se pri zagotavljanju in izboljševanju prevodov zanaša na umetno inteligenco. To velja tudi za funkcije, kot so samodejni podnapisi pri videoposnetkih.
- Pametni domovi, mesta in infrastruktura: Pametni termostati se učijo na podlagi našega vedenja, da prihranijo energijo (npr. zmanjšajo ogrevanje, ko nas ni doma). Snovalci pametnih mest upajo, da bo umetna inteligenca lahko pomagala urejati promet in s tem izboljšala povezljivost in zmanjšala prometne zastoje.
- Avtomobili: Avtonomna vozila še niso v splošni uporabi, a avtomobili že uporabljajo varnostne funkcije na podlagi umetne inteligence. EU je pomagala financirati avtomatske senzorje VI-DAS, ki zaznavajo možnost nevarnih situacij in nesreč.
- Navigacija temelji v veliki meri na umetni inteligenci.
- Kibernetska varnost: Sistemi umetne inteligence pomagajo prepoznati kibernetske napade in druge kibernetske grožnje ter se boriti proti njim, saj so zmožni procesirati stalni vnos podatkov, prepoznati vzorce in iskati izvor napadov.
- Boj proti koronavirusni bolezni 2019: V boju proti covidu-19 je bila umetna inteligenca uporabljena za termalne slike na letališčih in drugod. V medicini lahko pomaga prepoznati infekcije s slik, pridobljenih z računalniško tomografijo. Uporabljena je bila tudi za pridobivanje podatkov o širjenju bolezni.
- Boj proti dezinformacijam: Umetna inteligenca lahko prepozna lažne novice in dezinformacije, tako da med podatki na družbenih omrežjih išče senzacionalistične vsebine ter prepoznava, kateri spletni viri so verodostojni [2].
- Igre, umetnost, pisanje, glasba itd.

2.3.2 Možne uporabe umetne inteligence

- V zdravstvu: Raziskovalci preučujejo, kako uporabiti umetno inteligenco za analizo velikih količin zdravstvenih podatkov in iskanje vzorcev, ki bi pripeljali do novih odkritij v medicini. Razvili so program na podlagi umetne inteligence, ki se uporablja pri odzivanju na klice za nujno pomoč in naj bi prepoznal srčni zastoj pri klicatelju hitreje in točneje kot oseba v klicnem centru. Še en primer je KConnect, ki ga sofinancira Evropska unija. Razvija namreč večjezične storitve zagotavljanja vsebin in iskanja, ki ljudem pomagajo najti najrelevantnejše zdravstvene informacije.
- V prometu: Umetna inteligenca bi lahko pomagala izboljšati varnost, hitrost in učinkovitost železniškega prometa, tako da bi zmanjšala trenje, povečala hitrost in omogočila avtonomno vožnjo.
- V proizvodnji: Umetna inteligenca lahko pomaga proizvajalcem izboljšati učinkovitost in celo spodbudi tovarne, da se vrnejo nazaj v Evropo. Uporaba robotov v proizvodnji, optimiziranje prodajnih poti ali pravočasno predvidevanje okvar in vzdrževanja v pametnih tovarnah so med možnimi načini uporabe. Lahko prinese še večje prednosti: EU med drugim sofinancira raziskovalni projekt SatisFactory, ki uporablja sisteme, prilagojene resničnosti, za izboljšanje zadovoljstva delavcev v pametnih tovarnah.
- V kmetijstvu: Umetna inteligenca lahko pripomore k oblikovanju trajnostnega prehranskega sistema – z zmanjševanjem uporabe gnojil, pesticidov in herbicidov lahko zagotovi bolj zdravo hrano, pripomore pa k produktivnosti in zmanjša vplive na okolje. Roboti bi lahko denimo odstranjevali plevel, s čimer bi zmanjšali potrebo po herbicidih. Veliko kmetij v Evropski uniji že uporablja umetno inteligenco za avtomatsko in prilagojeno krmljenje živali ali za termostate v hlevih.
- V javni upravi in storitvah: Na podlagi široke palete razpoložljivih podatkov in prepoznavanja vzorcev bi umetna inteligenca lahko zgodaj opozorila na možnost naravne katastrofe ter s tem zagotovila učinkovito pripravo in blaženje posledic [2].

3 UMETNA INTELIGENCA V GLASBI

Na glasbenem področju lahko najdemo veliko primerov umetne inteligence in tudi principov, po katerih delujejo. Umetna inteligenca lahko deluje na osnovi surovega zvoka (npr. Jukebox) ali na osnovi glasbenih datotek, najpogosteje datotek MIDI (npr. MuseNet, AIVA, Magenta).

3.1 JUKEBOX

Jukebox je nevronska mreža, ki jo je ustvarilo podjetje OpenAI. Nevronska mreža Jukebox je sposobna ustvariti glasbo z vokalom in spremljavo, česar ostale umetne inteligence niso sposobne. Razlog za to sposobnost je nivo, na katerem Jukebox deluje. Deluje namreč na nivoju surovega zvoka, kar je prednost in tudi slabost. Prednost tega sistema je, da je z njim mogoče generirati tudi zvoke, ki niso na notnem črtovju, kot je npr. petje. Njegova slabost pa je, da zaradi povečane količine podatkov potrebuje več časa za generiranje pesmi.

Umetna inteligenca se za ustvarjanje glasbe uporablja že več kot pol stoletja. Eden najpomembnejših pristopov je ustvarjanje glasbe v obliki notacije, ki določa čas, višino, hitrost in inštrument vsake note, ki jo je treba igrati. To je privedlo do impresivnih rezultatov, kot je produkcija polifonične glasbe z več inštrumenti.

Tukaj se pojavi problem, da notacijski generatorji ne morejo ujeti človeških glasov in drugih kompleksnejših dinamik ter izraznosti, ki so bistvene za glasbo. Zato se je pojavil drugačen pristop za generiranje glasbe, in sicer z uporabo surovega zvoka. Uporaba surovega zvoka je zelo zahtevna, saj so zvočne sekvence zelo dolge (npr. tipična 4-minutna skladba CD-kakovosti [44 kHz, 16 bit] ima več kot 10 milijonov časovnih korakov). Za primerjavo – povprečna 4-minutna skladba, zapisana v datoteki MIDI, ima le 1500 časovnih korakov. Tako bi se moral model, da bi se naučil semantike glasbe na visoki ravni, soočiti z odvisnostmi znotraj izredno dolgega obsega.

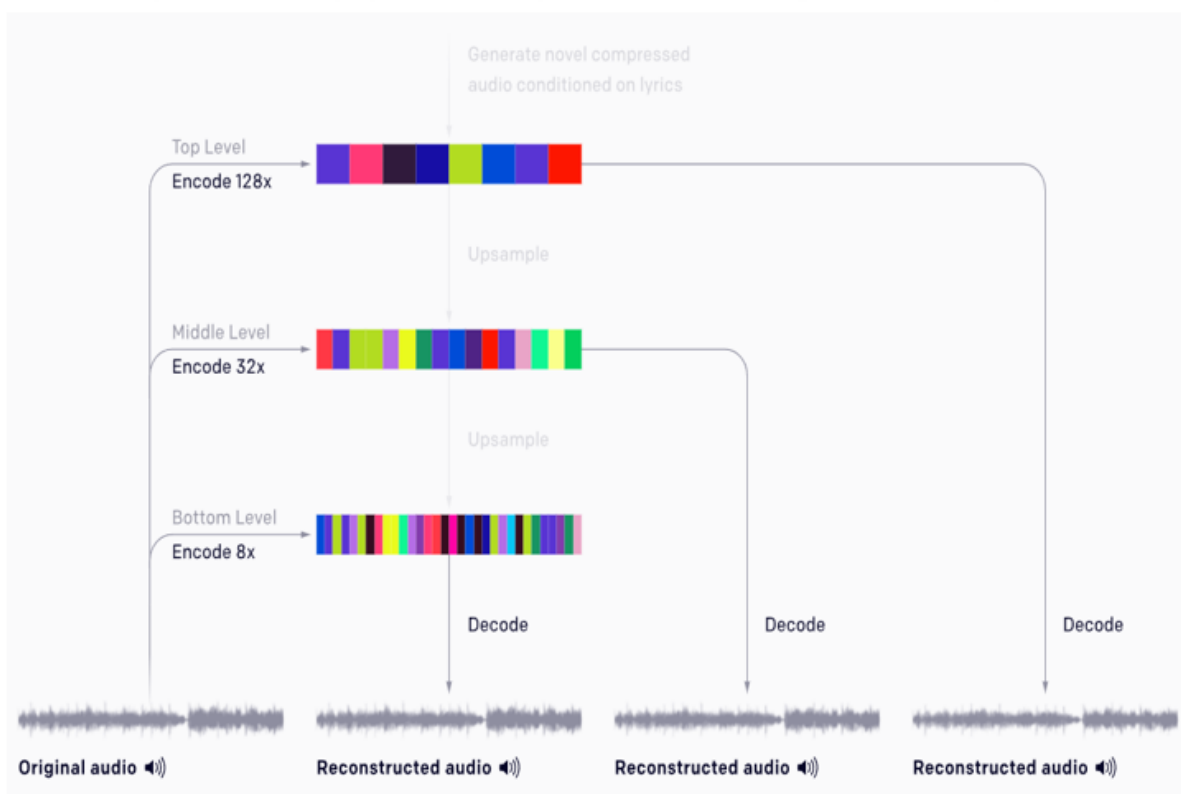
Eden od načinov za reševanje težave z ogromno količino podatkov je uporaba samodejnega kodirnika, ki stisne neobdelani zvok in zavrže zaznavno nepomembne bite informacij. Nato lahko uporabimo ta model za generiranje novega zvoka, ki ga na koncu pretvorimo nazaj v surovi zvok [8].

3.1.1 Stiskanje zvoka

Jukeboxov samodejni kodirnik za stiskanje zvoka deluje po principu kodirnika VQ-VAE. VQ-VAE ali daljše *Vector Quantized Variational AutoEncoder*, in sicer deluje po postopku vektorske kvantizacije.

V Jukeboxu se uporabljajo trije nivoji kodiranja VQ-VAE (slika 7), ki stisnejo 44 kHz surovi zvok (priloga 1) 8-krat (priloga 2), 32-krat (priloga 3) in 128-krat (priloga 4). Pri takšnem stiskanju zvoka se izgubi velik del zvočnih podrobnosti, to pa lahko slišimo kot hreščanje, ki je pri večjem stiskanju opaznejše. A vendar ohranja bistvene informacije o višini zvokov, zvenu in glasnosti zvoka [8].

Vsak nivo VQ-VAE kodira neodvisno glede na vhod. Najnižja raven kodiranja ustvari najkvalitetnejšo rekonstrukcijo, medtem ko najvišja raven kodiranja ohrani samo bistvene glasbene informacije.



Slika 7: Stiskanje zvoka [10]

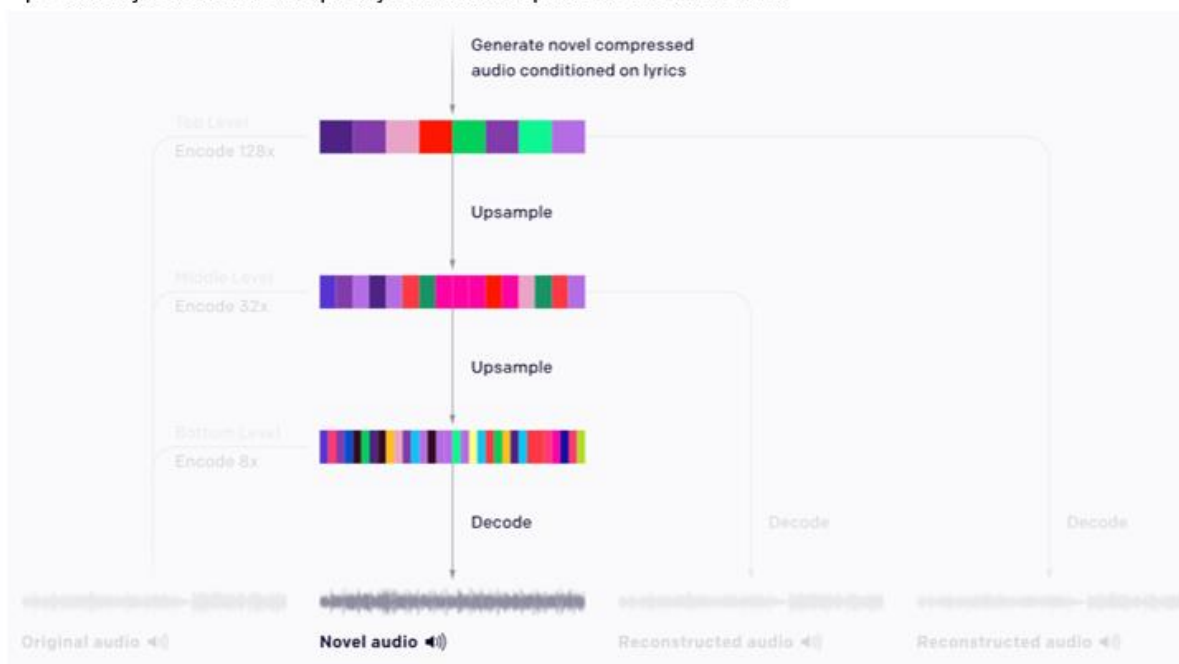
3.1.2 Generiranje zvoka

S prejšnjimi stisnjenimi modeli se naučimo pravila in jih nato uporabimo za ustvarjanje glasbe. Tako kot pri stiskanju zvoka imamo tri nivoje: prvi (najvišji) nivo generira najbolj stisnjen zvok, drugi in tretji nivo pa izboljšata kakovost zvoka (odpravita šumenje).

Najvišji nivoji modela modelirajo dolgotrajne strukture glasbe in vzorce. Dekodirani zvoki s te ravni imajo nižjo kakovost zvoka, vendar zajemajo semantiko na visoki ravni (petje in melodijo). Srednji in spodnji nivo pa dodata lokalne glasbene strukture, kot je npr. zven, kar znatno izboljša kakovost zvoka.

Te avtoregresivne modele treniramo s pomočjo Sparsovih transformatorjev. Ko so vsi modeli naučeni, lahko z njimi začnemo generirati nove zvoke, ki jih nato dekodiramo nazaj v surovi zvok (priloga 8) z uporabo dekoderja VQ-VAE [8].

Za ustvarjanje novih pesmi kaskada transformatorjev generira kode od zgornje do spodnje ravni, po kateri jih dekodeer na spodnji ravni lahko pretvori v surovi zvok.



Slika 8: Generiranje zvoka [6]

3.1.3 Nabor podatkov za učenje

Jukebox je treniran s pomočjo 1,2 milijona pesmi (od tega jih je 600.000 v angleščini), združenih z ustreznimi besedili in metapodatki. Metapodatki vključujejo izvajalca, zvrst albuma in leto objave pesmi, skupaj z razpoloženji in ključnimi besedami, povezanimi z vsako skladbo [8].

3.1.4 Besedilo

Pri treniranju Jukeboxa z besedilom se pojavi velik problem, kako povezati besedilo pesmi z glasbo, saj za dani kos zvoka ne moremo vedeti, ali se pojavi del besedila ali ne ali pa se različice pesmi ne ujemajo z različicami besedila ... To se lahko zgodi, če določeno skladbo izvaja več različnih izvajalcev na nekoliko različne načine. Poleg tega pevci pogosto ponavljajo fraze ali kako drugače spreminjajo besedila na načine, ki niso vedno zajeti v napisanih besedilih.

Da bi zvočne dele uskladili z ustreznimi besedili, začnemo s preprosto hevrstiko, ki poravna znake besedila tako, da linearno obsega trajanje vsake pesmi in prenese okno znakov fiksne velikosti, osredotočeno na trenutni segment med učenjem. Čeprav ta preprosta strategija linearne poravnave deluje presenetljivo dobro, ne uspe pri nekaterih žanrih s hitrimi besedili, kot je npr. hiphop. Da bi to rešili, uporabljamo Spleeter za ekstrakcijo vokala iz vsake pesmi in NUS AutoLyricsAlign na ekstrahiranih vokalih, da dobimo natančne poravnave besedila na ravni besed.

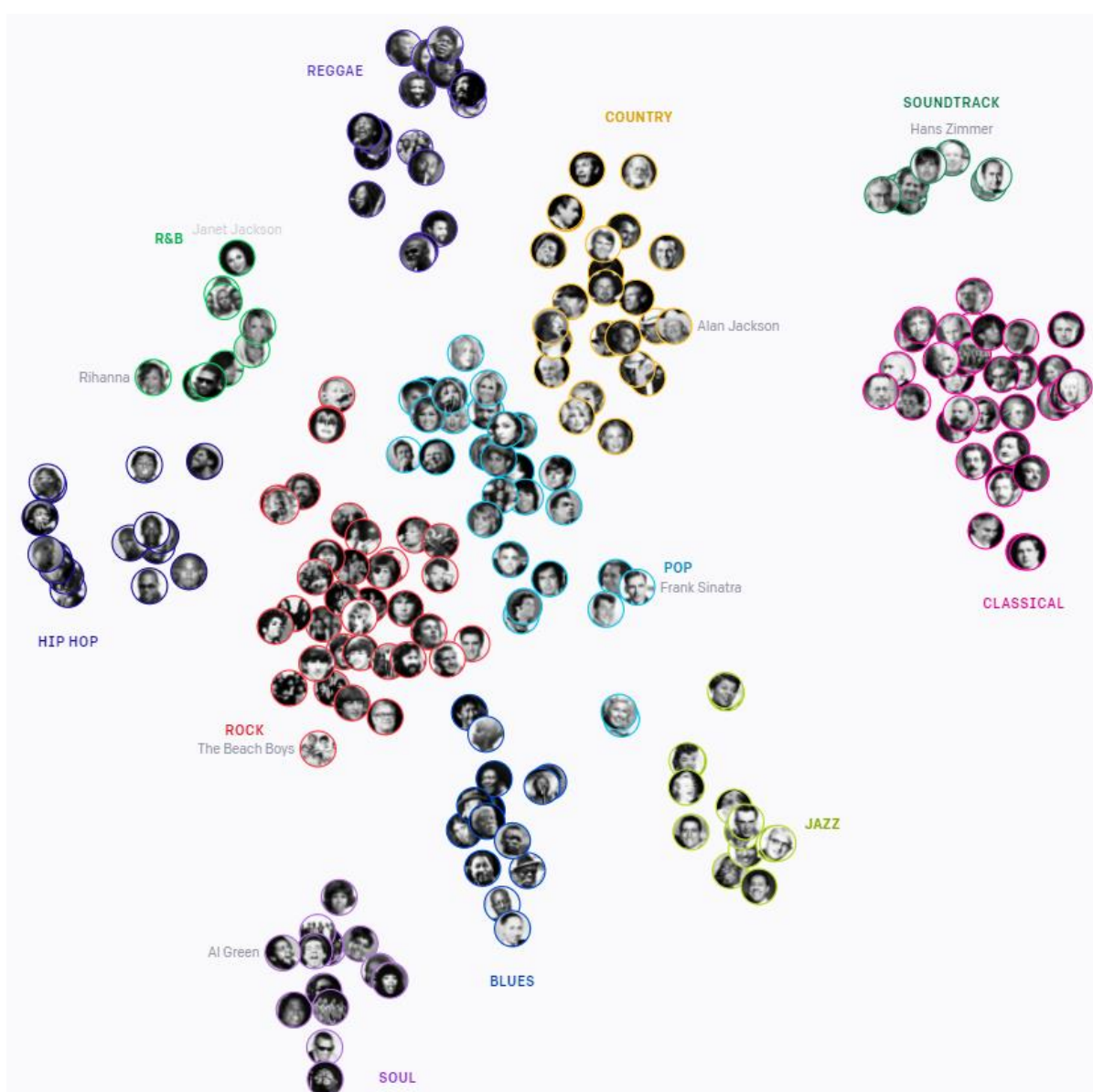
Za spremljanje besedil uporabljamo kodirnik (ustvari predstavitev besedila) z dodatnimi sloji pozornosti, ki uporabljajo poizvedbe iz glasbenega dekodirnika za spremljanje vrednosti iz kodirnika besedil. Po treningu se model nauči natančnejše poravnave [8].

3.1.5 Avtor in žanr

Za vsako skladbo Jukebox shranjuje dodatne informacije, kot sta podatka o izvajalcu in zvrsti. To ima dve prednosti:

1. Model je sposoben doseči boljšo kakovost v katerem koli določenem slogu.
2. V času generacije lahko model usmerjamo, da ustvarja v slogu, ki ga izberemo.

Slika 9 prikazuje, kako se model nauči na nenadzorovani način združiti podobne izvajalce in zvrsti tesno skupaj [8].



Slika 9: Razvrščanje izvajalcev po žanrih [9]

3.1.6 Omejitve

Medtem ko Jukebox predstavlja korak naprej v glasbeni kakovosti, skladnosti, dolžini zvočnega vzorca in zmožnosti pogojevanja z izvajalcem, žanrom in besedilom, obstaja velika vrzel med temi generacijami in glasbo, ki jo ustvarijo ljudje.

Npr. medtem ko ustvarjene pesmi kažejo lokalno glasbeno skladnost, sledijo tradicionalnim vzorcem akordov in lahko vsebujejo celo impresivne sole, ne slišimo večjih znanih glasbenih struktur, kot so refreni, ki se ponavljajo. Naš postopek zmanjšanja in povečanja vzorčenja uvaja opazen šum. Izboljšanje VQ-VAE, tako da njegove kode zajemajo več glasbenih informacij, bi lahko to zmanjšale. Naši modeli se prav tako počasi vzorčijo zaradi avtoregresivne narave vzorčenja. Popolno upodabljanje minute zvoka prek naših modelov traja približno 9 ur, zato jih še ni mogoče uporabiti v interaktivnih aplikacijah [8].

3.2 MUSENET

MuseNet je globoka nevronska mreža, ki lahko generira 4-minutne skladbe z desetimi različnimi inštrumenti in zmožna združevati sloge od Mozarta do Beatlov. MuseNet je poseben zato, ker ni izrecno programiran z našim razumevanjem glasbe, ampak je sam odkril vzorce harmonije, ritma in sloga. Tako se je naučil predvideti nadaljevanje pesmi na podlagi 100.000 datotek MIDI.

Ker MuseNet pozna veliko različnih stilov, lahko generirano glasbo združimo na nove načine. V tem primeru model dobi prvih šest not Chopinove skladbe Nocturne, vendar želimo, da ustvari skladbo v pop slogu s klavirjem, bobni, basom in kitaro. Model uspe prepričljivo združiti oba sloga [7].

3.2.1 Omejitve MuseNeta

- Inštrumenti, ki jih izberemo, so samo predlogi, ne zahteve. MuseNet ustvari vsako noto z izračunom verjetnosti za vse možne note in inštrumente. Model se premakne, da bo naša izbira inštrumenta verjetnejša, vendar vedno obstaja možnost, da bo izbral nekaj drugega.
- MuseNet težje povezuje nenavadne pare slogov in inštrumentov (npr. Chopin z basom in bobni). Generacije bodo naravnejše, če izberemo inštrumente, ki so najbližji običajnemu slogu skladatelja ali skupine [7].

3.2.2 Delovanje MuseNeta

Programerji so pridobili podatke za učenje MuseNeta iz različnih virov. Transformator, ki so ga usposobili za zaporedne podatke, so prosili, da je iz nabora not napovedal prihajajočo noto, nakar so poskušali eksperimentirati z več različnimi načini kodiranja datotek MIDI v žetone, in sicer tako, da so uporabili pristop po akordih, v katerem so v prvem pristopu skušali kombinacijo not, ki zvenijo naenkrat, obravnavati kot posamezen akord ter vsakemu akordu dodati žeton. V drugem pristopu pa so glasbene vzorce zgostili tako, da so se osredotočili samo na začetke not in jih skušali stisniti z uporabo sheme kodiranja bajtnih parov.

Tudi za označevanje poteka časa so uporabili dve različni metodi, in sicer žetone, ki so bili skalirani glede na tempo glasbe (ti so predstavljali glasbeni utrip ali delček utripa), in žetone, ki so absolutni čas označevali v sekundah.

Po izvedbi so se odločili za kodiranje, ki združuje v en sam žeton tako izraznost kot jedrnatost, in sicer v višini, glasnosti in inštrumentu [7].

4 GENERIRANJE GLASBE S POMOČJO MAGENTE

4.1 IDEJA

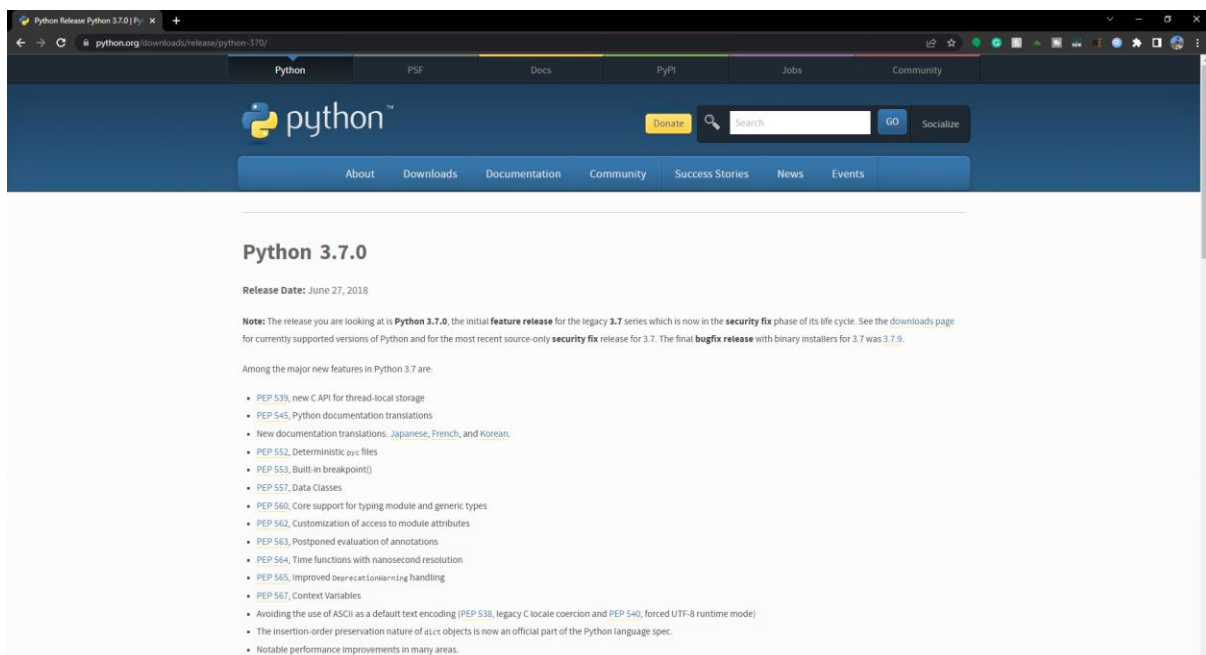
Najprej sem se odločil, da bom ustvaril model, s katerim bi lahko generirali glasbo, nakar pa sem si zastavil nalogo, da bom za uresničitev cilja uporabil knjižnico Magenta, ki ponuja precej umetne inteligence za ustvarjanje glasbe.

4.2 ODLOČANJE

Med raziskovanjem knjižnice Magenta sem opazil veliko različnih vrst generatorjev, ki so se zelo razlikovali glede na kompleksnost učenja in generiranja. Zaradi omejene računalniške moči sem se odločil, da ne bom uporabil boljših generatorjev. Tako sem iz svoje izbire izločil sete `score2perf`, `melody_rnn`, `coconet` itd. Tako mi je ostal `performance_rnn`.

4.3 PRIPRAVA DELOVNEGA OKOLJA

Najprej sem s spleta prenesel zadnjo verzijo Pythona¹, ki jo knjižnica Magenta omogoča, torej verzijo 3.7.

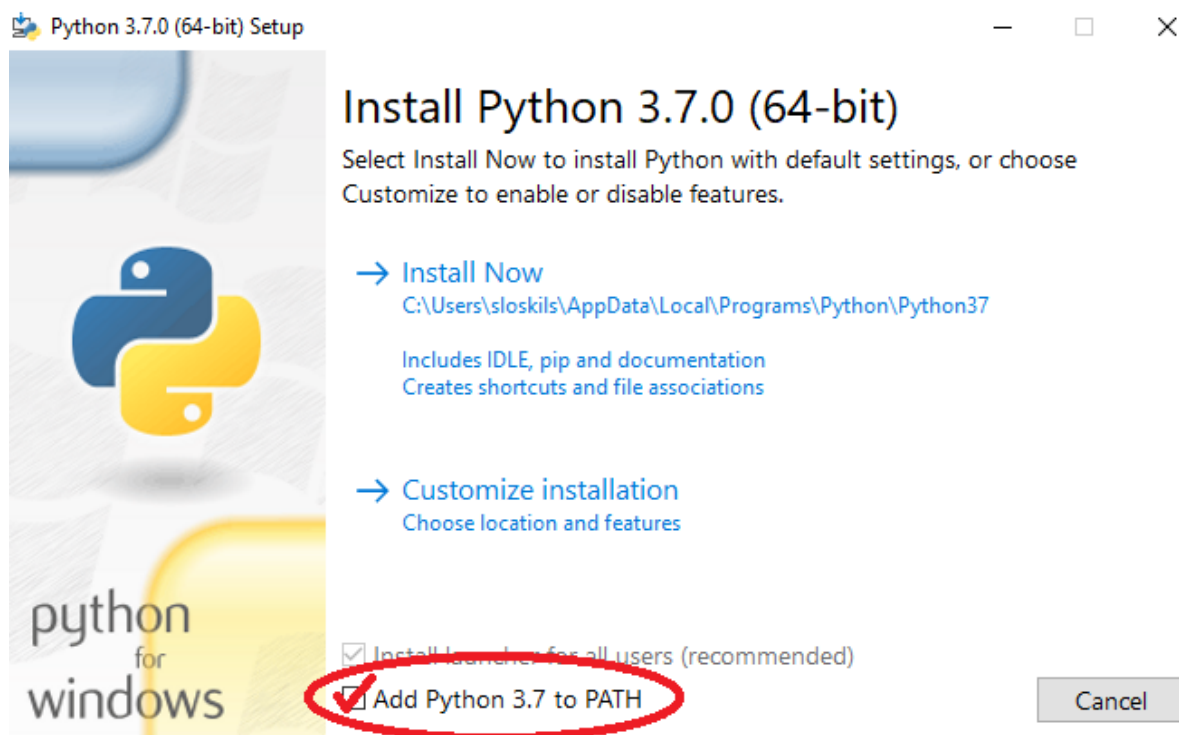


Slika 10: Pythonova spletna stran

(Vir: osebni arhiv)

¹ Python (<https://www.python.org/downloads/release/python-370>, 20. 2. 2022)

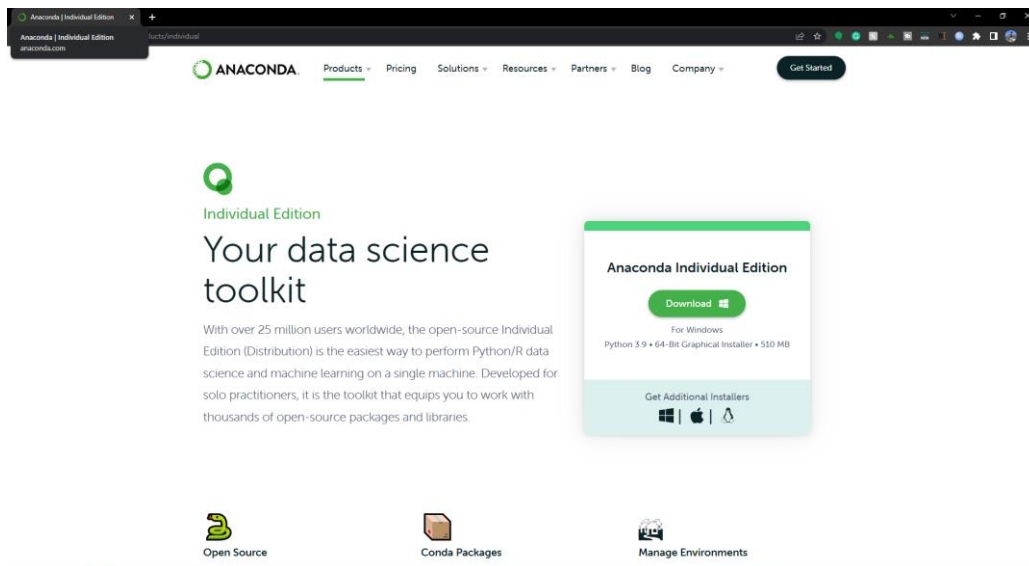
Preden namestimo Python, ga moramo dodati v PATH. To omogoča, da bomo lahko priklicali Python iz ukaznega poziva.



Slika 11: Namestitev programa Python

(Vir: osebni arhiv)

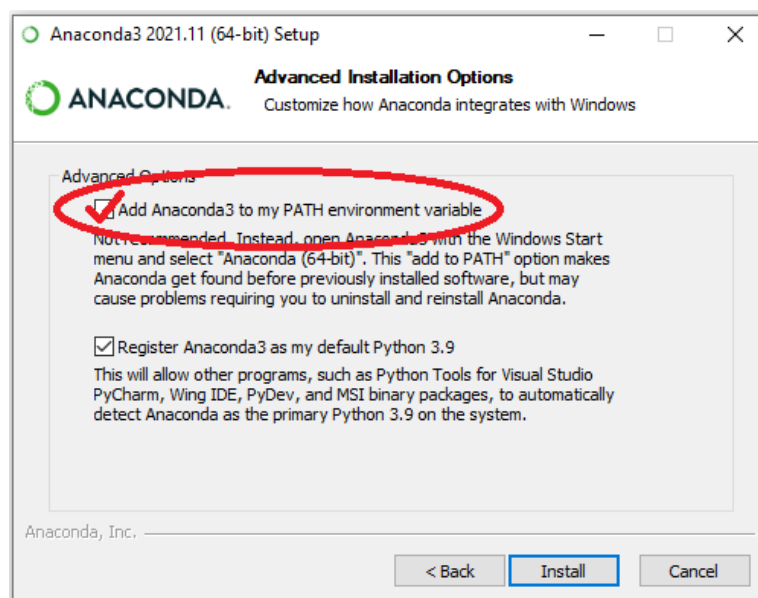
Nato lahko namestimo orodje Anaconda², ki omogoča ustvarjanje okolij, ki lahko uporabljajo različne verzije Pythona in preprečijo mešanje knjižnic med projekti.



Slika 12: Spletna stran orodja Anaconda

(Vir: osebni arhiv)

Preden zaključimo z namestitvijo, predlagam, da tudi orodje Anaconda dodamo v PATH, ker nam bo to olajšalo preklapljanje med okolji.

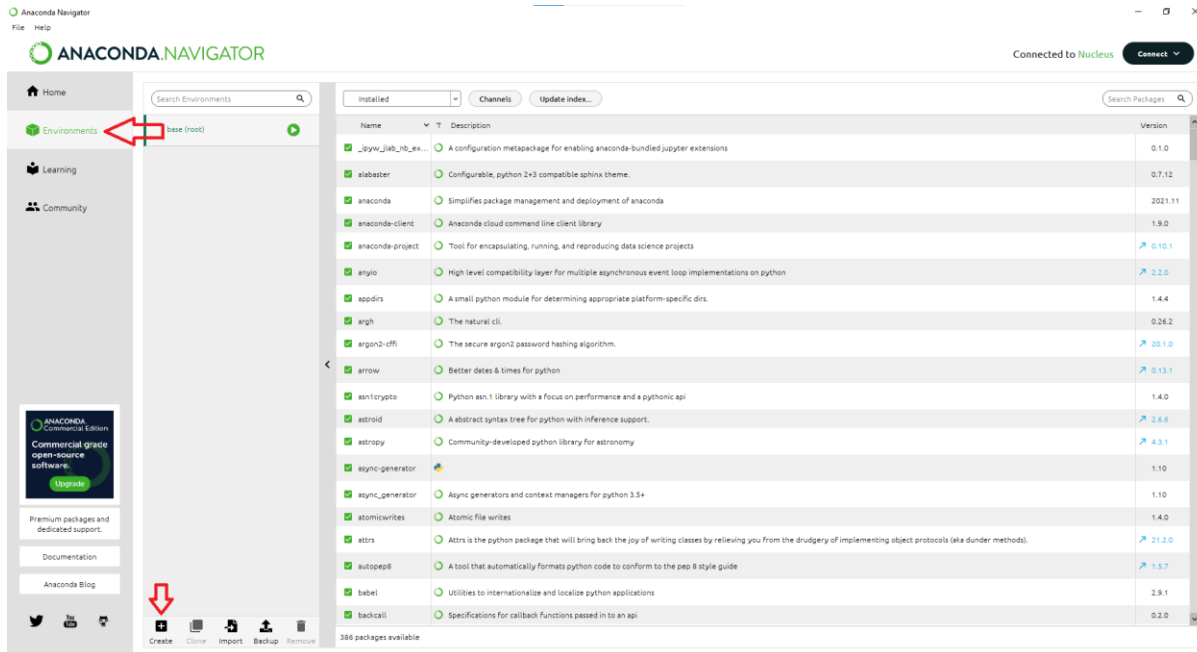


Slika 13: Anaconda (namestitev)

(Vir: osebni arhiv)

² Anaconda (<https://www.anaconda.com/products/individual>, 20. 2. 2022)

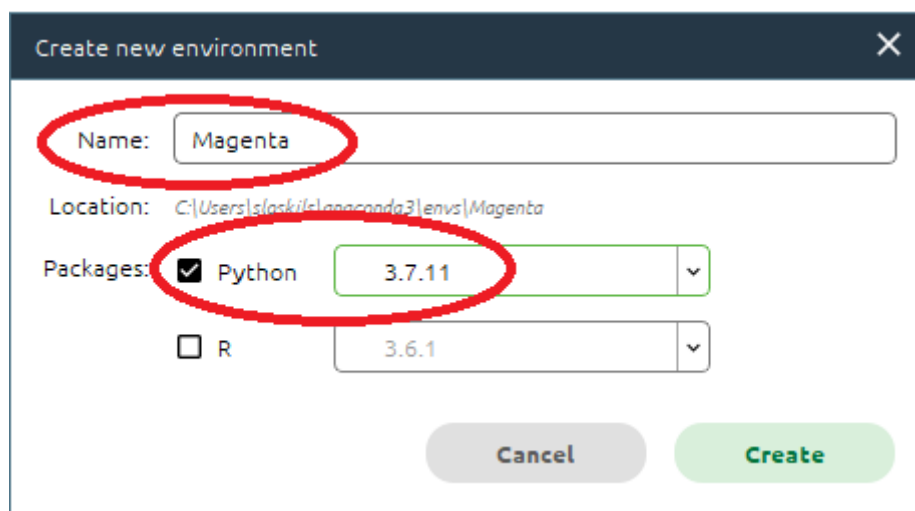
Po uspešni namestitvi zaženemo *Anaconda Navigator (anaconda3)*. Najprej v levem izbirnem meniju izberemo *Environments*, nato desno spodaj izberemo *Create*.



Slika 14: Ustvarjanje novega programskega okolja

(Vir: osebni arhiv)

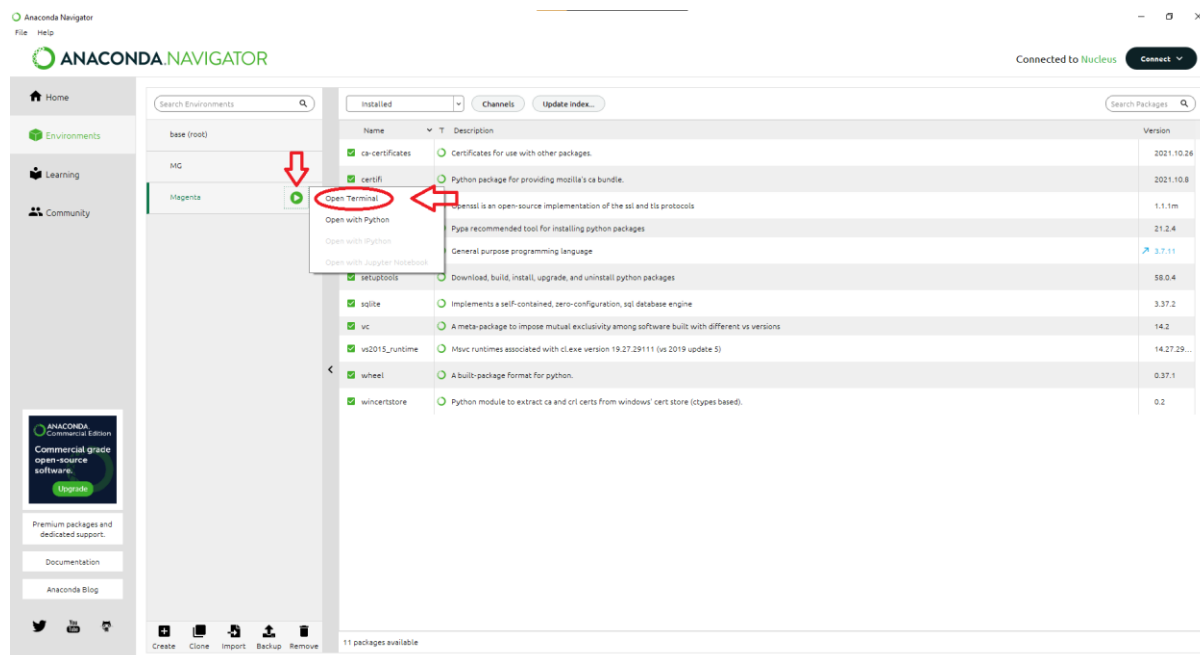
V pojavno okno vpišemo ime novega okolja in izberemo programski jezik Python, verzijo 3.7.11. Za ta korak potrebujemo dostop do interneta.



Slika 15: Izbira parametrov za novo okolje

(Vir: osebni arhiv)

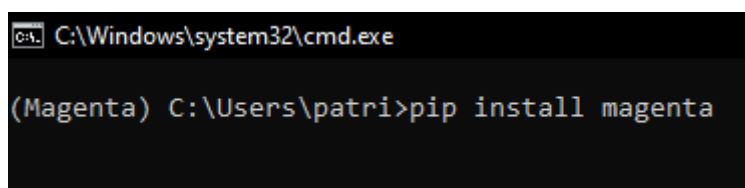
Ko smo ustvarili novo okolje, lahko v njem zaženemo ukazni poziv (angl. *command prompt*) oz. terminal. To naredimo tako, da kliknemo na zeleni krožec z belim trikotnikom in v prikazanem seznamu izberemo *Open Terminal*.



Slika 16: Odpiranje terminala znotraj novega okolja

(Vir: osebni arhiv)

Odpre se črno okno, v katerem sta v oklepaju označena okolje, v katerem delamo, in pot do mape, v kateri se nahajamo. Za namestitev knjižnice uporabimo ukaz »*pip install magenta*« in zahtevo s klikom potrdimo. Za ta korak potrebujemo dostop do interneta.



Slika 17: Ukaz za namestitev knjižnice Magenta

(Vir: osebni arhiv)

Začne se izpisovati ogromna količina vrstic, ki nas samo obveščajo o trenutnem koraku programa. Okno pustimo odprto, dokler se ponovno ne izpiše vrstica, ki smo jo videli na začetku. S tem ukazom smo zaključili nastavljanje okolja.

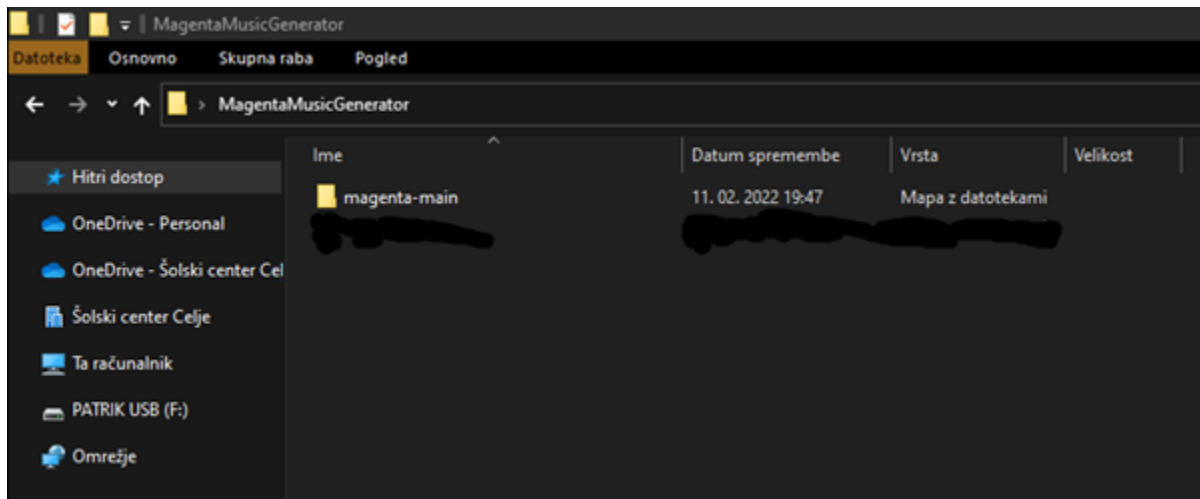
```
C:\Windows\system32\cmd.exe
Using cached tensorflow_metadata-1.6.0-py3-none-any.whl (48 kB)
Collecting tensorflow-hub>=0.2
Using cached tensorflow_hub-0.12.0-py2.py3-none-any.whl (108 kB)
Building wheels for collected packages: gym, mir-eval
Building wheel for gym (PEP 517) ... done
Created wheel for gym: filename=gym-0.22.0-py3-none-any.whl size=708396 sha256=452678dd592df8e26bff5e2ed691107e7e31a339d197822886b89d
Stored in directory: c:\users\patri\appdata\local\pip\cache\wheels\7d\5e\87\7d50e0179edda70feff5bba05c381041e1c1fd80c6b06a4cc3
Building wheel for mir-eval (setup.py) ... done
Created wheel for mir-eval: filename=mir_eval-0.7-py3-none-any.whl size=100721 sha256=f301286cf0c767349be2cc9e8d0663a8d28261fa41a3b40
Stored in directory: c:\users\patri\appdata\local\pip\cache\wheels\18\5a\46\d2527fff1fd975e1a793375e6ed763bfe4d3ea396b7cdc470eb
Successfully built gym mir-eval
Installing collected packages: urllib3, pyasn1, idna, charset-normalizer, zipp, typing-extensions, six, rsa, requests, pyparsing, pycpa5
tl, scipy, requests-oauthlib, parso, numba, MarkupSafe, joblib, importlib-metadata, httplib2, gym-notices, googleapis-common-protos, go
emplate, typeguard, tqdm, tornado, termcolor, tensorflow-probability, tensorflow-metadata, tensorflow-hub, tensorboard-plugin-wit, tens
, pygments, prompt-toolkit, promise, Pillow, pickleshare, packaging, opencv-python, mpmath, mido, matplotlib-inline, markdown, Jinja2, 5
lib2, google-api-core, gin-config, future, dill, click, cached-property, backcall, audioread, wrapt, tiffiffle, tf-slim, tf-estimator-ni
rd, tabulate, sympy, PyWavelets, pypng, pydub, pretty-midi, pandas, opt-einsum, oauth2client, networkx, mesh-tensorflow, librosa, libcl
gle-pasta, google-api-python-client, gevent, gast, fonttools, flatbuffers, flask, dopamine-rl, dm-tree, cycler, bz2file, bokeh, attrs, r
mir-eval, matplotlib, dm-sonnet, magenta
Successfully installed IPython-7.31.1 Jinja2-3.0.3 MarkupSafe-2.1.0 Pillow-9.0.1 PyWavelets-1.2.0 PyYAML-6.0 Werkzeug-2.0.3 absl-py-1.0o
ty-1.5.2 cachetools-5.0.0 cffi-1.15.0 charset-normalizer-2.0.12 click-8.0.4 cloudpickle-2.0.0 colorama-0.4.4 cycler-0.11.0 decorator-5.0.
.29.1 future-0.18.2 gast-0.5.3 gevent-21.12.0 gin-config-0.5.0 google-api-core-2.5.0 google-api-python-client-2.37.0 google-auth-2.6.0 g
.0 greenlet-1.1.2 grpcio-1.44.0 gunicorn-20.1.0 gym-0.22.0 gym-notices-0.0.4 h5py-3.6.0 httplib2-0.20.4 idna-3.3 imageio-2.16.0 importla
1.1.0 keras-2.8.0 keras-preprocessing-1.1.2 kfac-0.2.0 kiwisolver-1.3.2 libclang-13.0.0 librosa-0.7.2 llvmlite-0.32.1 magenta-2.1.3 mara
math-1.2.1 networkx-2.6.3 note-seq-0.0.3 numba-0.49.1 numpy-1.21.5 oauth2client-4.1.3 oauthlib-3.2.0 opencv-python-4.5.5.62 opt-einsum-
oolkit-3.0.28 protobuf-3.19.4 pyasn1-0.4.8 pyasn1-modules-0.2.8 pycparser-2.21 pydub-0.25.1 pygments-2.11.2 pygtrie-2.4.2 pyparsing-3.0.
b-1.3.1 resampy-0.2.2 rsa-4.8 scikit-image-0.19.2 scikit-learn-1.0.2 scipy-1.7.3 six-1.16.0 sk-video-1.1.10 sortedcontainers-2.4.0 sounl
rd-data-server-0.6.1 tensorboard-plugin-wit-1.8.1 tensorflow-2.8.0 tensorflow-addons-0.16.1 tensorflow-datasets-4.5.2 tensorflow-gan-2.g
bability-0.7.0 termcolor-1.1.0 tf-estimator-nightly-2.8.0.dev20211122109 tf-slim-1.1.0 threadpoolctl-3.1.0 tiffiffle-2021.11.2 tornado-6.1
8 wcmwidth-0.2.5 wrapt-1.13.3 zipp-3.7.0 zope.event-4.5.0 zope.interface-5.4.0
(Magenta) C:\Users\patri>
```

Slika 18: Izpis terminala po uspešni namestitvi Magente

(Vir: osebni arhiv)

4.4 PRIPRAVA PODATKOV ZA UČENJE

Po nastavljenem okolju si na trdem disku ustvarimo mapo, v kateri bomo hranili celoten projekt. Nato s spleta prenesemo Githubovo knjižnico Magenta³. Preneseno datoteko ZIP nato premaknemo v našo delovno mapo in jo razširimo. Po razširjanju lahko arhivsko datoteko izbrišemo. Datoteke iz te mape nam bodo razjasnile, kako uporabljati knjižnico.



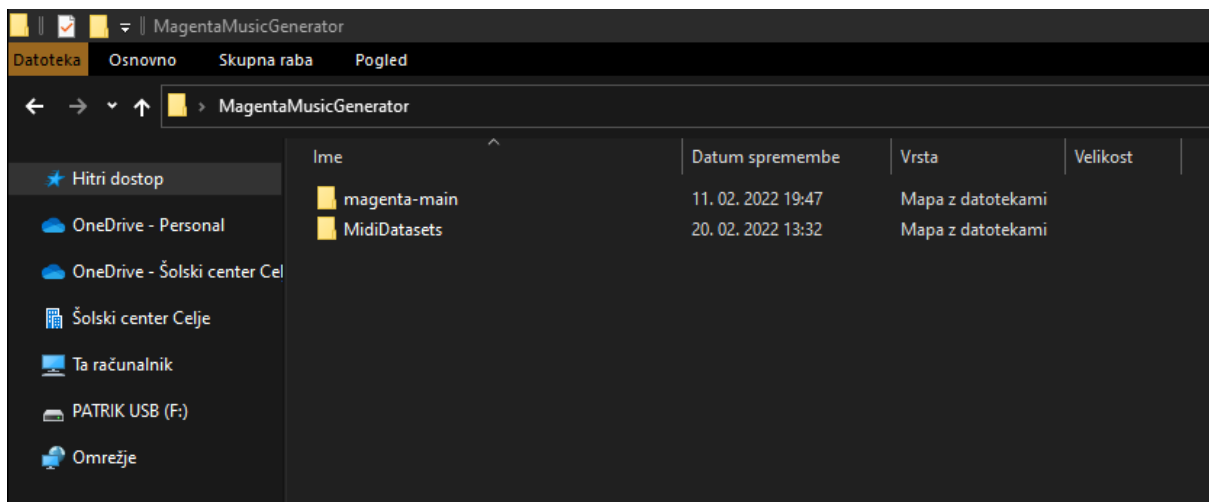
Slika 19: Izgled delovne mape po dodajanju knjižnice

(Vir: osebni arhiv)

Po pripravi programskega dela potrebujemo datoteke MIDI, s katerimi bomo trenirali AI. Za ta primer bomo uporabili zbirko datotek MIDI Johanna Sebastiana Bacha⁴. Najprej bomo znotraj naše delovne mape ustvarili novo mapo, v kateri bomo imeli vse datoteke MIDI, s katerimi bomo učili model (sliki 20 in 21).

³ Github Magenta (<https://github.com/magenta/magenta>, 20. 2. 2022)

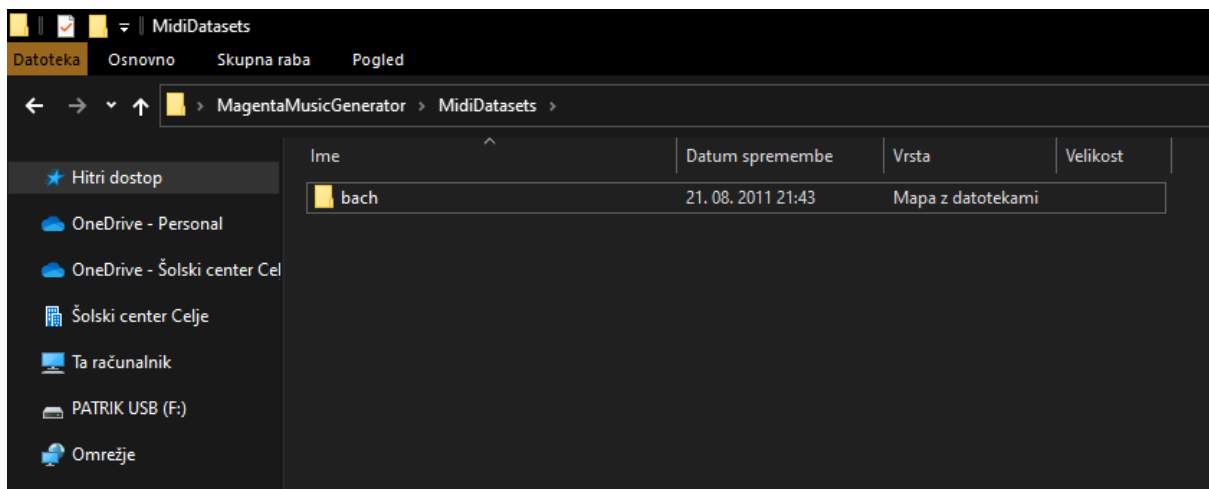
⁴ Bachova zbirka (<http://www.bachcentral.com/MIDIindex.html>, 20. 2. 2022)



Slika 20: Izgled delovne mape

(Vir: osebni arhiv)

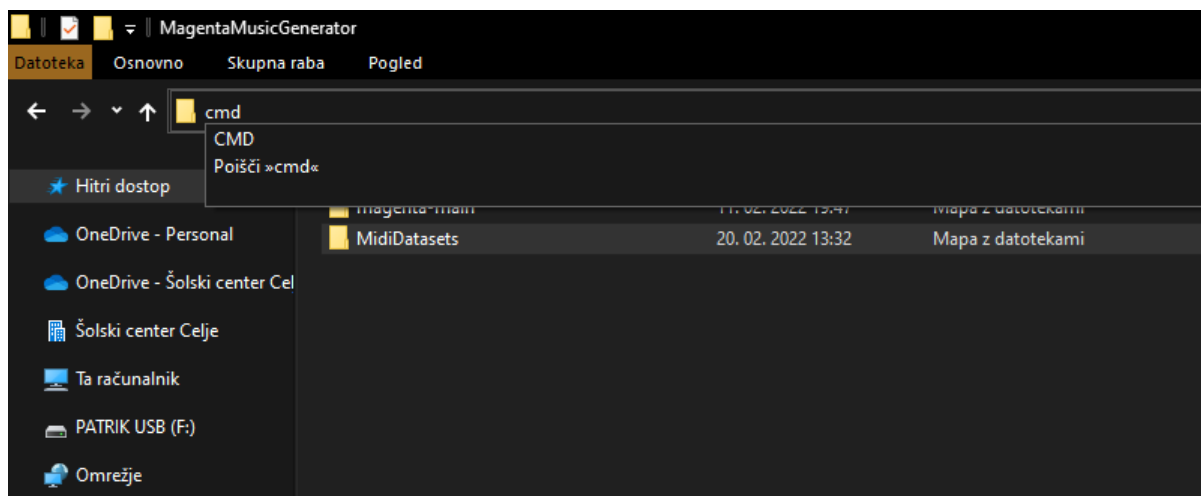
Zbirko datotek MIDI prenesemo, jo premaknemo v novo mapo in jo razširimo. Izbrišemo vse odvečne datoteke. Tako nam ostane samo mapa »bach«.



Slika 21: Izgled delovne mape

(Vir: osebni arhiv)

Sedaj se vrnemo v delovno mapo in iz nje zaženemo ukazni poziv. To naredimo tako, da v polje, v katerem je zapisana pot do mape, napišemo ukaz »cmd« in pritisnemo enter.



Slika 22: Odpiranje terminala iz mape

(Vir: osebni arhiv)

Znova se nam odpre črno okno, ampak tokrat lahko opazimo, da pred potjo ni v oklepaju zapisanega okolja, ki smo ga pripravili. Zato ga moramo priklicati z ukazom »conda activate magenta«. Sedaj lahko vidimo v oklepaju zapisano ime okolja in pot do naše delovne mape.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. Vse pravice pridržane.

C:\Users\patri\Desktop\MagentaMusicGenerator>conda activate magenta
(magenta) C:\Users\patri\Desktop\MagentaMusicGenerator>
```

Slika 23: Aktivacija programskega okolja

(Vir: osebni arhiv)

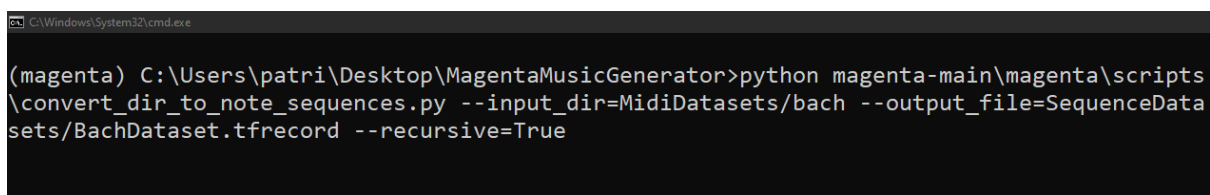
Ko se nahajamo v našem okolju in delovni mapi, lahko začnemo s pretvarjanjem datotek MIDI v »note sequence« datoteko. To naredimo z ukazom:

```
»python magenta-main\magenta\scripts\convert_dir_to_note_sequences.py /
--input_dir=MIDIDatasets/bach /
--output_file=SequenceDatasets/BachDataset.tfrecord /
--recursive=True«.
```

Beseda »python« na začetku pove ukaznemu pozivu, da bomo zagnali datoteko s pomočjo Pythona. Sledi pot do programa, ki ga želimo zagnati. Zatem z »--« določimo parametre:

- a) `--input_dir` določi mesto, kjer se nahajajo datoteke MIDI;
- b) `--output_file` določi mesto in ime »note sequence« datoteke, ki jo ustvarjamo;
- c) `--recursive` pove, ali želimo uporabiti tudi datoteke MIDI, ki se nahajajo v podmapah.

Ta program mora delovati neprekinjeno. Če se program prekine sredi dela, bomo morali ustvarjeno datoteko izbrisati in program znova zagnati. Čas delovanja programa je odvisen od tehničnih lastnosti računalnika in količine datotek MIDI. Paziti moramo tudi na prostor na trdem disku, saj je lahko nova datoteka tudi štiri- ali večkrat večja od izvirne.



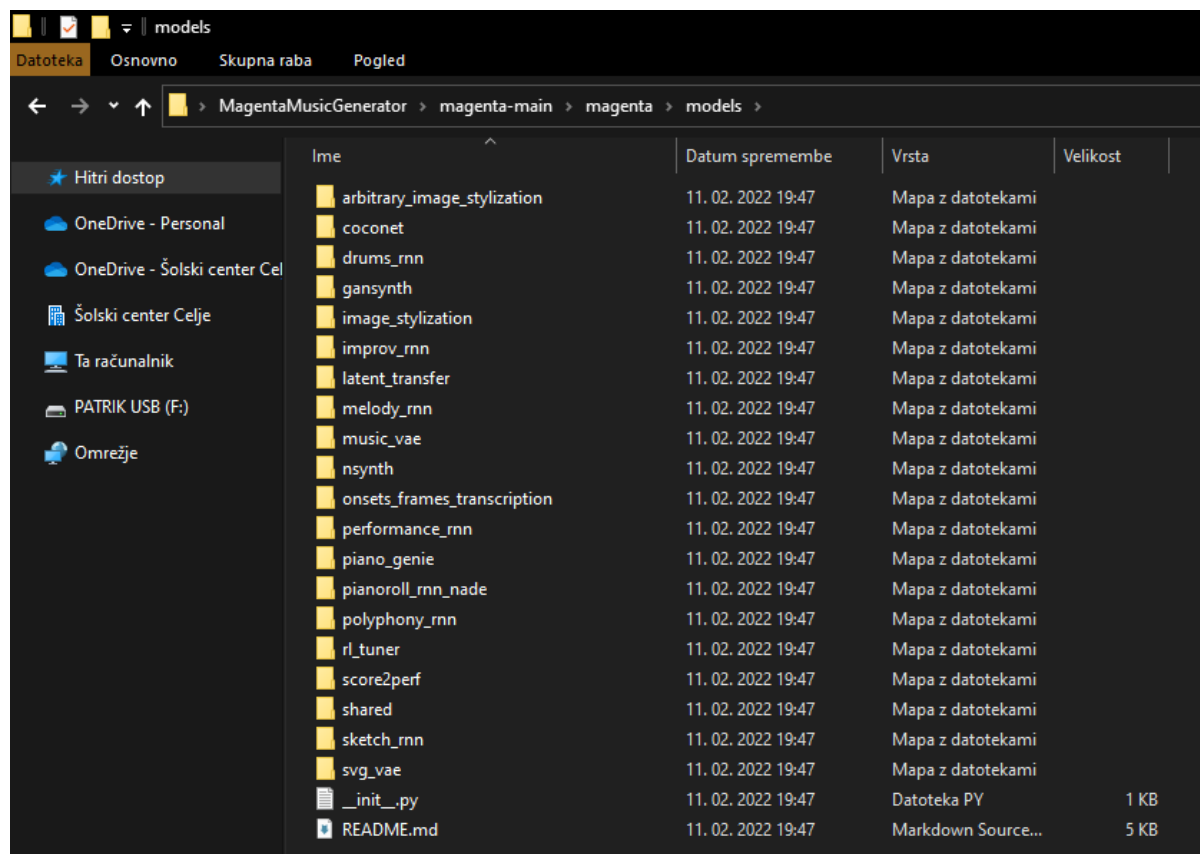
```
C:\Windows\System32\cmd.exe
(magenta) C:\Users\patri\Desktop\MagentaMusicGenerator>python magenta-main\magenta\scripts
\convert_dir_to_note_sequences.py --input_dir=MidiDatasets/bach --output_file=SequenceData
sets/BachDataset.tfrecord --recursive=True
```

Slika 24: Ukaz za pretvorbo datotek MIDI v »note sequence« datoteko

(Vir: osebni arhiv)

4.5 TRENIRANJE MODELA IN GENERIRANJE GLASBE

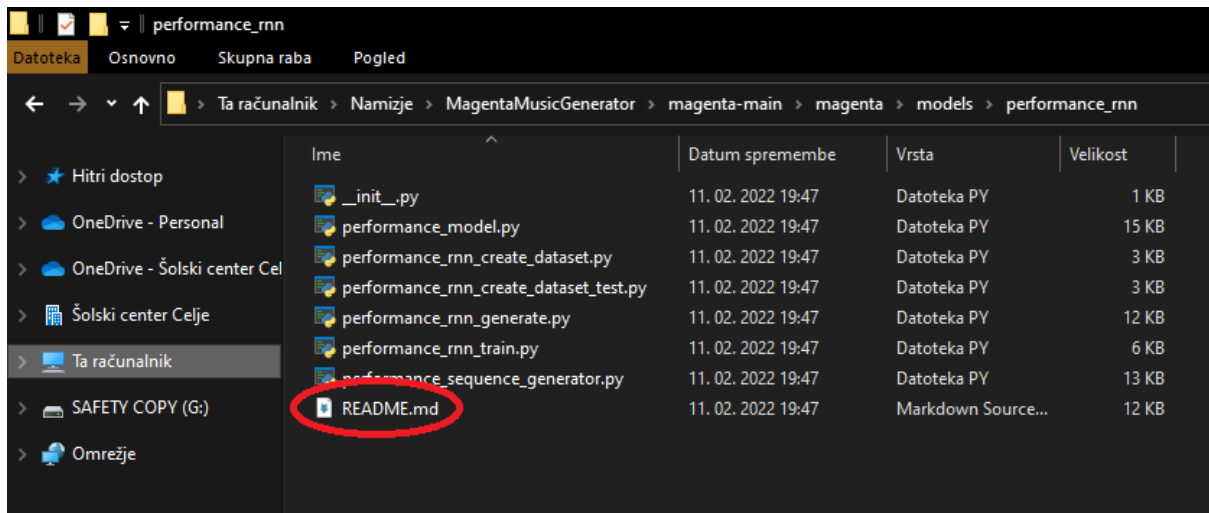
Ko program deluje, se moramo odločiti, katero umetno inteligenco bomo uporabili za ustvarjanje svojega modela. Vse možnosti umetne inteligence lahko najdemo znotraj svoje delovne mape, in sicer znotraj »magenta-main/magenta/models«.



Slika 25: Vse možnosti umetne inteligence znotraj knjižnice Magenta

(Vir: osebni arhiv)

Slika 25 prikazuje različne možnosti izbire umetne inteligence. Sam sem se odločil, da bom uporabil *performance_rnn*. V mapi *performance_rnn* odpremo datoteko *README.md*. V njej lahko najdemo vse informacije o uporabi umetne inteligence.



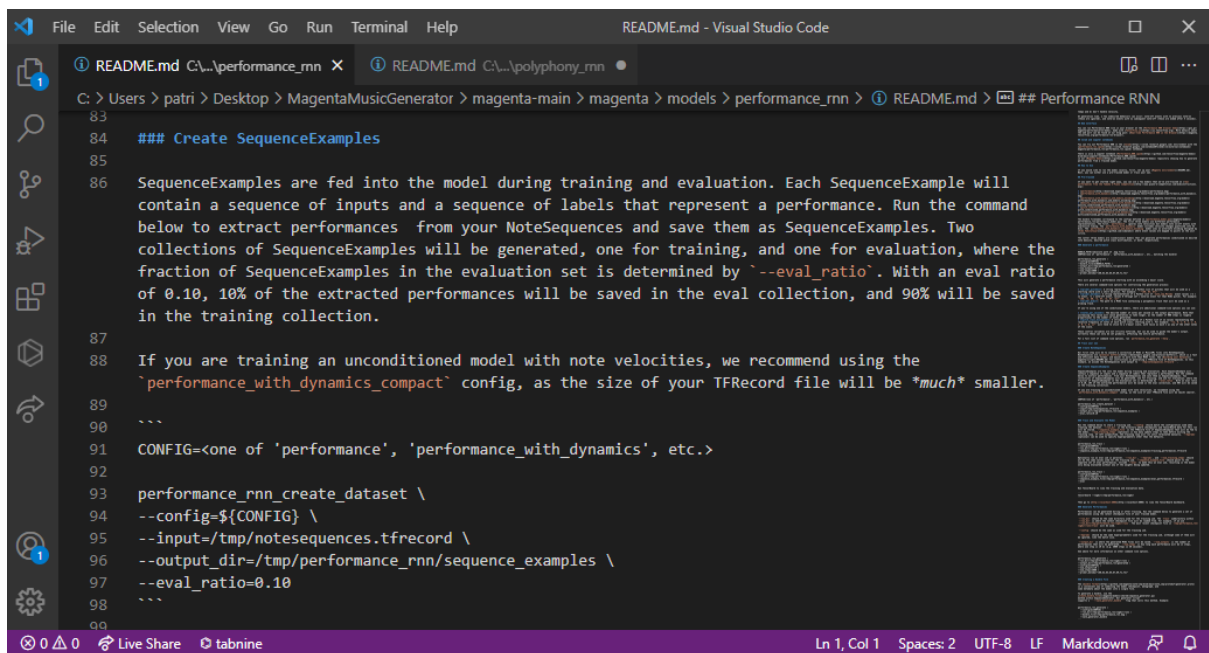
Slika 26: Opiranje datoteke README.md

(Vir: osebni arhiv)

V datoteki README.md najdemo ukaz za ustvarjanje nove podatkovne baze za učenje:

```
»performance_rnn_create_dataset
--config=performance_with_dynamics
--input=SequenceDatasets/BachDataset.tfrecord
--output_dir=SequenceExamples/BachPerformance_rnn
--eval_ratio=0.10«.
```

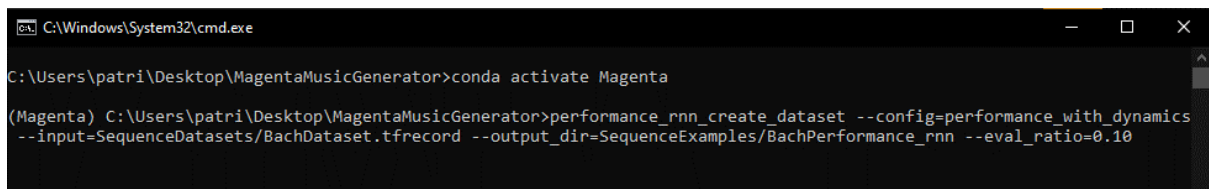
Kopirano kodo prilagodimo, da se sklada s strukturo shranjevanja datotek, prav tako izberemo tudi `--config`, ki je v mojem primeru `performance_with_dynamics`.



Slika 27: Datoteka README.md

(Vir: osebni arhiv)

Novo kodo zažene v ukaznem pozivu.



Slika 28: Ukaz v terminalu znotraj delovne mape in programskega okolja

(Vir: osebni arhiv)

Ko program zaključi z ustvarjanjem nove podatkovne baze, lahko začnemo s treniranjem modela. Uporabimo ukaz:

```
»performance_rnn_train
--config=performance_with_dynamics
--run_dir=LogDir/BachPerformance_rnn
--sequence_example_file=SequenceExamples/BachPerformance_rnn/
training_performances.tfrecord«.
```



```
C:\Windows\System32\cmd.exe
C:\Users\patri\Desktop\MagentaMusicGenerator>conda activate Magenta
(Magenta) C:\Users\patri\Desktop\MagentaMusicGenerator>performance_rnn_train --config=performance_with_dynamics --run_dir=LogDir/BachPerformance_rnn --sequence_example_file=SequenceExamples/BachPerformance_rnn/training_performances.tfrecord
```

Slika 29: Ukaz za treniranje modela

(Vir: osebni arhiv)

Treniranje modela traja relativno dolgo, če želimo dobre rezultate. Prav zato sem za treniranje svojega modela porabil teden dni. Postopek treniranja lahko nadzorujemo, tako da v novem ukaznem pozivu znotraj okolja in delovne mape zaženemo ukaz:

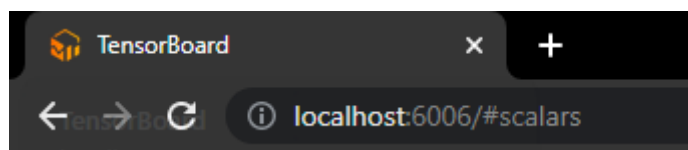
»`tensorboard --logdir=LogDir/BachPerformance_rnn`«.

```
C:\Windows\System32\cmd.exe - tensorboard --logdir=LogDir/BachPerformance_rnn
C:\Users\patri\Desktop\MagentaMusicGenerator>conda activate Magenta
(Magenta) C:\Users\patri\Desktop\MagentaMusicGenerator>tensorboard --logdir=LogDir/BachPerformance_rnn
2022-02-27 13:51:02.415118: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudnn64_8.dll'; dLError: cudnn64_8.dll not found
2022-02-27 13:51:02.415266: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1850] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are installed properly if you would like to use GPU. Follow the guide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
Serving TensorBoard on localhost; to expose to the network, use a proxy or pass --bind_all
TensorBoard 2.8.0 at http://localhost:6006/ (Press CTRL+C to quit)
W0227 13:55:07.869416 34036 plugin_event_accumulator.py:319] Found more than one graph event per run, or there was a metagraph containing a graph_def, as well as one or more graph events. Overwriting the graph with the newest event.
W0227 13:55:07.886422 34036 plugin_event_accumulator.py:331] Found more than one metagraph event per run. Overwriting the metagraph with the newest event.
```

Slika 30: Ukaz in izpis terminala za nadzorovanje treniranja

(Vir: osebni arhiv)

Nato v brskalnik napišemo »`http://localhost:6006/`« in izrišejo se nam vsi grafi in podatki o napredku učenja modela.



Slika 31: URL v brskalniku

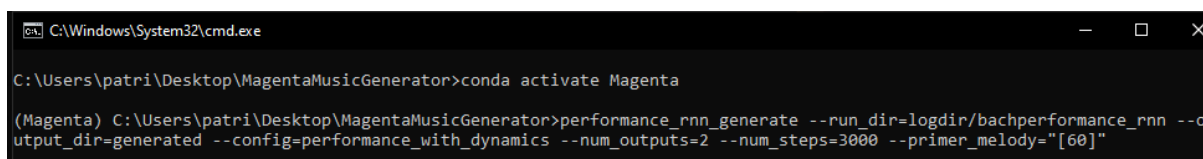
(Vir: osebni arhiv)

Ko menimo, da je model natreniran, lahko poskusimo generirati nove skladbe. To naredimo z ukazom:

```
»performance_rnn_generate --run_dir=logdir/bachperformance_rnn --  
output_dir=generated --config=performance_with_dynamics --num_outputs=2 --  
num_steps=3000 --primer_melody="[60]"«.
```

Če želimo za osnovo uporabiti datoteko MIDI, lahko to storimo tako, da zamenjamo »--primer_melody=« z »--primer_MIDI=« in nadaljujemo s potjo do želene datoteke.

Za generacijo pesmi pa lahko uporabimo tudi »bundle file«, to pa lahko storimo tako da zamenjamo »--run_dir=« z »--bundle-file« in nadaljujemo s potjo do želene datoteke (priloga 15).

A screenshot of a Windows command prompt window. The title bar reads "C:\Windows\System32\cmd.exe". The prompt shows the user has navigated to "C:\Users\patri\Desktop\MagentaMusicGenerator" and executed "conda activate Magenta". The subsequent command is "performance_rnn_generate --run_dir=logdir/bachperformance_rnn --output_dir=generated --config=performance_with_dynamics --num_outputs=2 --num_steps=3000 --primer_melody="[60]"".

```
C:\Windows\System32\cmd.exe  
C:\Users\patri\Desktop\MagentaMusicGenerator>conda activate Magenta  
(Magenta) C:\Users\patri\Desktop\MagentaMusicGenerator>performance_rnn_generate --run_dir=logdir/bachperformance_rnn --o  
utput_dir=generated --config=performance_with_dynamics --num_outputs=2 --num_steps=3000 --primer_melody="[60]"
```

Slika 32: Ukaz za generiranje pesmi

(Vir: osebni arhiv)

4.6 UGOTOVITVE

4.6.1 Izbiranje podatkov za učenje

Pri izdelovanju modela sem naletel na problem, ki ga je povzročil nabor datotek MIDI. Na začetku sem menil, da ni pomembno, ali zberem nabor glasbe z različnimi žanri in avtorji ter svoj model treniram na tej osnovi. Po dveh dneh neprestanega treniranja modela sem se odločil, da bom ustvaril nekaj vzorcev, ki pa niso zveneli najboljše. Takrat sem menil, da je problem v dejstvu, da svojega modela nisem dovolj izuril. Tako sem se odločil počakati še tri dni in postopek sem ponovil (priloga). Takrat sem ugotovil, da ni dobro mešati preveč različnih žanrov med sabo. Odločil sem se, da bom izbral samo enega avtorja in gradil na tem. Novi model je bil izrazito boljši od starega.

4.6.2 Izbiranje umetne inteligence

Pri izbiranju in preizkušanju vrst umetne inteligence sem ugotovil, da so te med seboj zelo različne. Nekatere se hitro izurijo, ne potrebujejo toliko računalniške moči in ustvarjajo slabšo glasbo (npr. *melody_rnn* in *piano_genie*), medtem ko druge ustvarjajo boljšo glasbo, a potrebujejo veliko več računalniške moči (npr. *performance_rnn* in *score2perf*).

5 MNENJE DIJAKOV O UMETNI INTELIGENCI

5.1 ANKETA

Anketo je rešilo 60 naključnih dijakov Srednje šole za strojništvo, mehatroniko in medije.

Najprej sem se odločil, da anketirance povprašam, ali so že slišali za umetno inteligenco. 80 % vprašanih je odgovorilo, da so že slišali zanj. Odločil sem se, da nima smisla izbirati podatke med ljudmi, ki še nikoli niso slišali za umetno inteligenco, saj je celotno anketo izpolnilo samo 48 od 60 vprašanih.



Grafikon 1: Ali ste že slišali za umetno inteligenco?

(Vir: osebni arhiv)

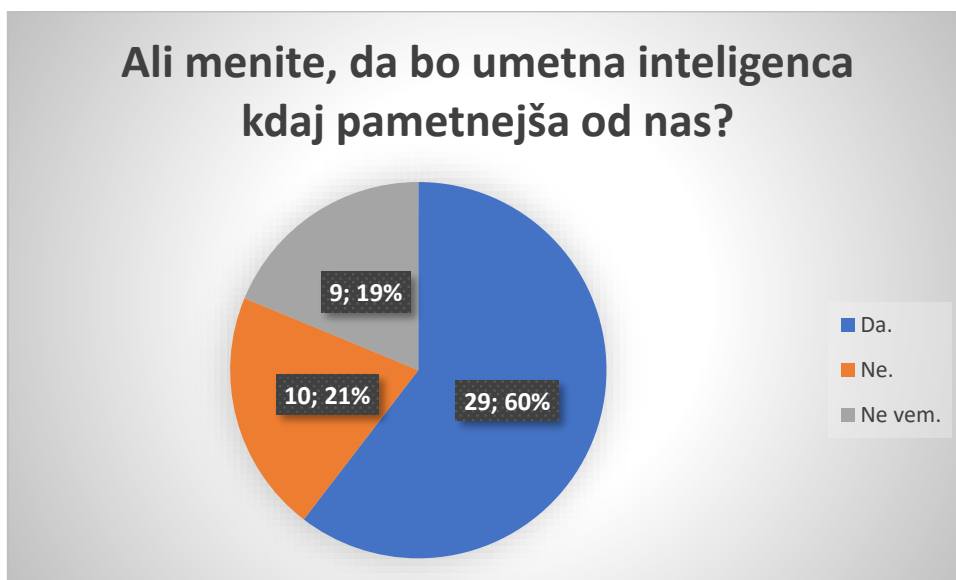
Na vprašanje, ali se umetna inteligenca razvija hitro, je velika večina odgovorila pritrdilno, kar se zdi razumljivo, saj je celotno področje umetne inteligence še vedno zelo novo.



Grafikon 2: Ali mislite, da se umetna inteligenca hitro razvija?

(Vir: osebni arhiv)

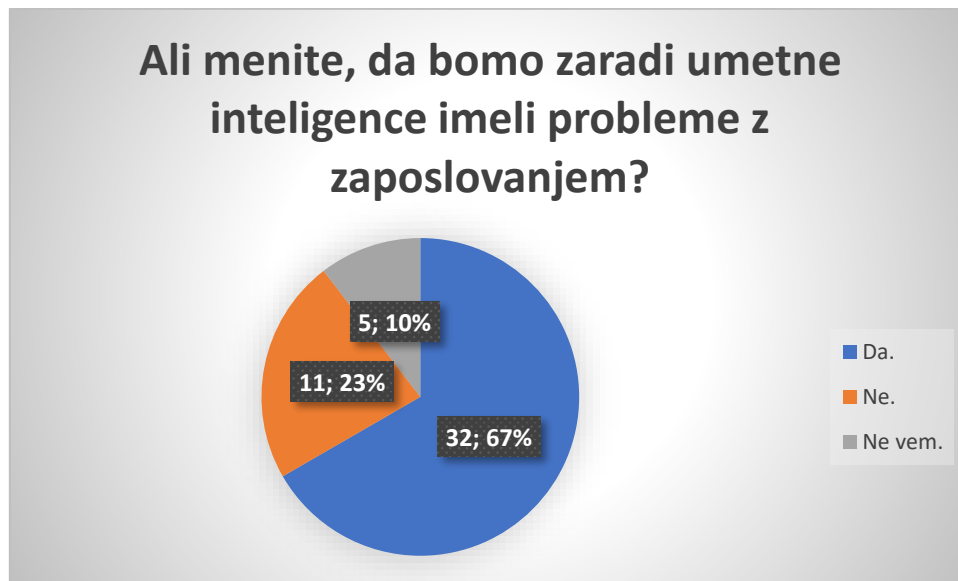
Sledilo je vprašanje, ali dijaki menijo, da bo umetna inteligenca kdaj pametnejša od človeštva. Na to vprašanje je kar 60,4 % vprašanih odgovorilo pritrdilno.



Grafikon 3: Ali menite, da bo umetna inteligenca kdaj pametnejša od nas?

(Vir: osebni arhiv)

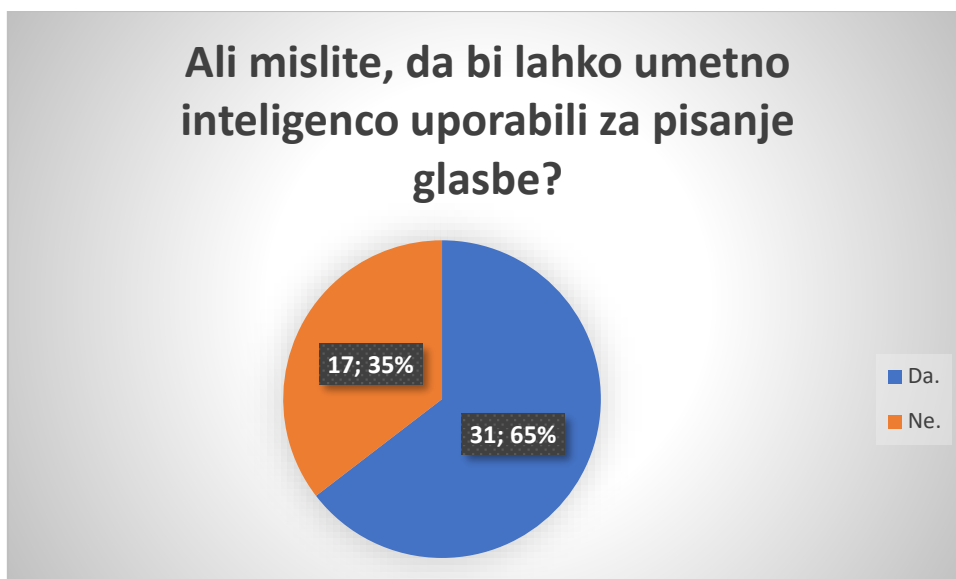
Na vprašanje o tem, ali bomo zaradi umetne inteligence imeli probleme z zaposlovanjem, je 66,7 % vprašanih odgovorilo, da se jim to zdi verjetno.



Grafikon 4: Ali menite, da bomo zaradi umetne inteligence imeli probleme z zaposlovanjem?

(Vir: osebni arhiv)

Sledilo je vprašanje o srečevanju z umetno inteligenco.



Grafikon 6: Ali mislite, da bi lahko umetno inteligenco uporabili za pisanje glasbe?

(Vir: osebni arhiv)

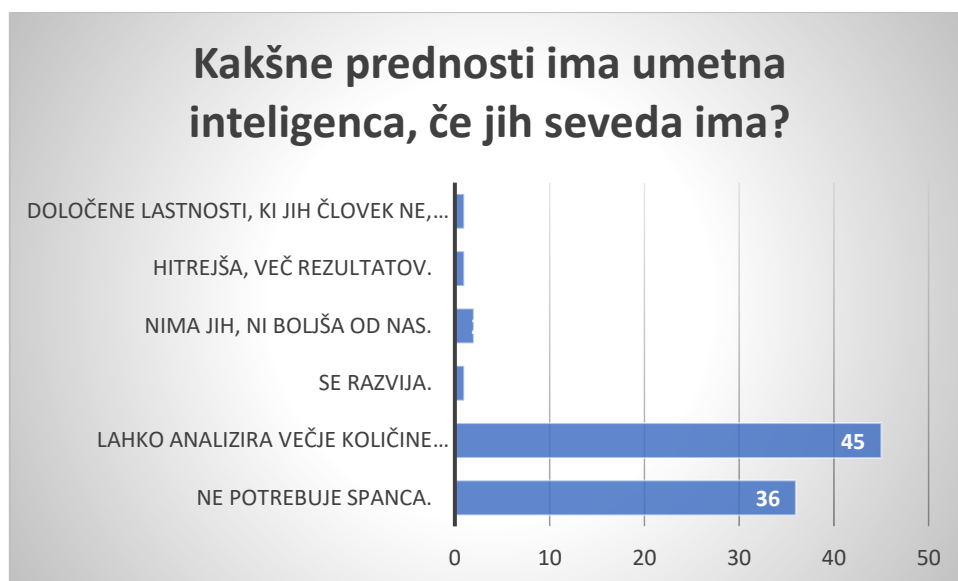
Na vprašanje o tem, ali bi lahko umetna inteligenca ustvarjala boljšo glasbo od nas, je z ne odgovorilo 81,3 % vprašanih, kar se mi zdi razumljivo, saj sem bil tudi sam na začetku takšnega mnenja. Odgovore na to trditev sem ovrgel s kvizom (Grafikon 11).



Grafikon 7: Ali mislite, da bi lahko umetna inteligenca ustvarjala boljšo glasbo od nas?

(Vir: osebni arhiv)

Kot zanimivost sem se odločil, da anketirance povprašam, kakšne prednosti ima umetna inteligenca. Ugotovil sem, da je po mnenju večine njena glavna prednost, da lahko analizira velike količine podatkov.



Grafikon 8: Kakšne prednosti ima umetna inteligenca, če jih seveda ima?

(Vir: osebni arhiv)

Pritrdilni odgovori na vprašanje o tem, ali že obstaja umetna inteligenca za ustvarjanje glasbe, so me presenetili, saj je njihov odstotek višji kot pri vprašanju o tem, ali bi lahko uporabili umetno inteligenco za pisanje glasbe.



Grafikon 9: Ali mislite, da že obstaja umetna inteligenca za ustvarjanje glasbe?

(Vir: osebni arhiv)

Odgovori na zadnje vprašanje so presenetljivi, saj je večina odgovorila, da bi poslušala glasbo, ki jo je napisala umetna inteligenca.



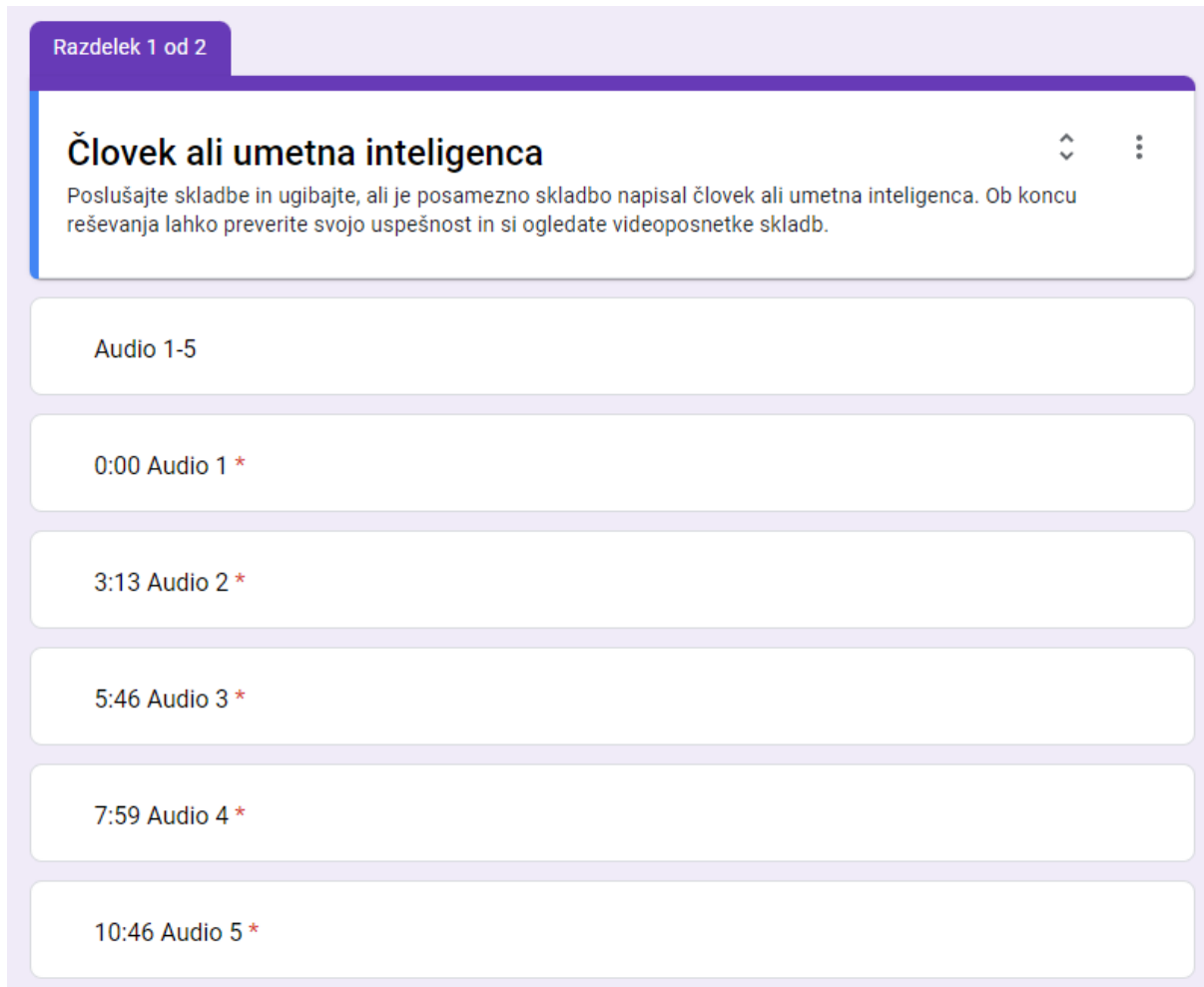
Grafikon 10: Ali bi poslušali glasbo, ki jo je napisala umetna inteligenca?

(Vir: osebni arhiv)

5.2 KVIZ

Kot zanimivost sem se odločil, da vsem, ki so odgovorili na prvo anketo, zastavim izziv.

Sestavil sem kviz iz desetih pesmi, iz petih »umetnih« in petih »neumetnih«.



The image shows a screenshot of a quiz interface. At the top, there is a purple header with the text 'Razdelek 1 od 2'. Below this, the main title of the quiz is 'Človek ali umetna inteligenca'. Underneath the title, there is a brief instruction: 'Poslušajte skladbe in ugibajte, ali je posamezno skladbo napisal človek ali umetna inteligenca. Ob koncu reševanja lahko preverite svojo uspešnost in si ogledate videoposnetke skladb.' The quiz consists of five audio clips, each with a timestamp and a label: 'Audio 1-5', '0:00 Audio 1 *', '3:13 Audio 2 *', '5:46 Audio 3 *', '7:59 Audio 4 *', and '10:46 Audio 5 *'. The interface is clean and modern, with a light purple and white color scheme.

Slika 33: Kviz

(Vir: osebni arhiv)

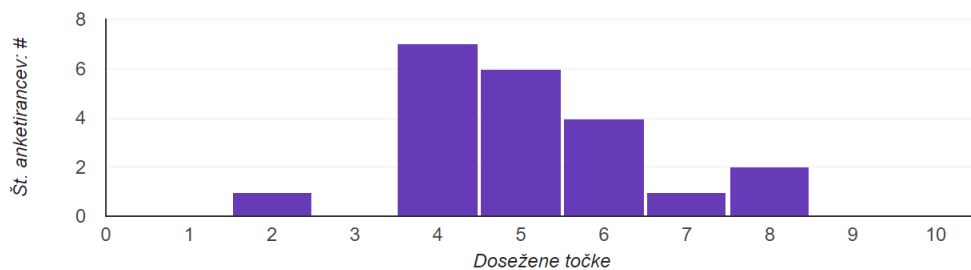
Na kviz je odgovorilo 21 oseb, ki so ugibale, ali je pesem napisal človek ali jo je generirala umetna inteligenca. Rezultati kviza me niso presenetili, saj sem tudi sam zelo težko ločeval med »umetno« glasbo in »neumetno«.

Povprečje
5,1/10 točk

Srednja vrednost
5/10 točk

Obseg
2–8 točk

Porazdelitev skupnega števila točk



Grafikon 11: Rezultati kviza

(Vir: osebni arhiv)

6 HIPOTEZE

V celotni raziskovalni nalogi sem potrdil naslednje hipoteze:

- ✓ HIPOTEZA 1: Umetna inteligenca lahko ustvarja glasbo.
- ✓ HIPOTEZA 2: Spletne strani za generiranje glasbe obstajajo.
- ✓ HIPOTEZA 4: Možno je uporabiti predustvarjene modele za generiranje glasbe.

Prvo in drugo hipotezo sem potrdil, ko sem našel generatorja melodij AIVA⁵ in MuseNet⁶, ki omogočata generiranje pesmi v različnih stilih in z različnimi inštrumenti.

Potrdil sem tudi četrto hipotezo, in sicer z odkritjem odprtokodne knjižnice Magenta.

Dve hipotezi pa sem ovrgel:

- ✗ HIPOTEZA 3: Ustvarjanje umetne inteligence za generiranje glasbe je preprosto.
- ✗ HIPOTEZA 5: Umetna inteligenca ne more generirati petja.

Tretjo hipotezo sem ovrgel po izobraževanju o grajenju umetne inteligence⁷, saj je pojasnilo vse ovire, ki jih je treba premagati, da pridemo do zelenega cilja.

Prav tako pa sem ovrgel tudi peto hipotezo, saj sem med svojim raziskovanjem naletel na umetno inteligenco z imenom Jukebox⁸, ki lahko generira melodije z besedilom.

⁵ Generator glasbe AIVA (<https://www.aiva.ai/>, 4. 3. 2022)

⁶ Generator glasbe MuseNet (<https://openai.com/blog/musenet/>, 4. 3. 2022)

⁷ Izobraževanje Building AI (<https://buildingai.elementsofai.com/>, 20. 2. 2022)

⁸ Generator glasbe Jukebox (<https://openai.com/blog/jukebox/>, 20. 2. 2022)

7 ZAKLJUČEK

Na začetku se mi je ideja o ustvarjanju lastne umetne inteligence zdela zelo preprosta, a pokazalo se je, da umetna inteligenca ni navaden program, ki ga lahko spišemo v nekaj mesecih. Umetna inteligenca je polna algoritmov in zank, ki jih s srednješolsko matematiko in logiko ni mogoče rešiti.

Tako sem se odločil, da bom namesto ustvarjanja svoje umetne inteligence uporabil že ustvarjeno in z njo izuril svoj model, ki ga bom nato uporabil za generiranje. Takrat sem naletel na Pythonovo knjižnico Magenta, za katero sem mislil, da bo rešila vse moje probleme. A z novimi rešitvami prihajajo tudi novi problemi. Pri nameščanju knjižnice sem se prvič srečal z orodjem Anaconda in programskimi okolji ter se poučil o njihovi uporabnosti.

Uporaba umetne inteligence v glasbi ima veliko prednosti, kot npr. ne potrebuje inspiracije za pisanje in lahko bolje poustvarja sloge raznih skladateljev. Ima pa tudi slabosti, kot so, da potrebuje ogromno računalniške moči za učenje in ne more tako dobro ločevati čustev ter jih vpeljevati v melodije.

Menim, da je za umetno inteligenco na področju glasbe naslednji najbolj logični korak implementacija čustev ali da bi lahko umetna inteligenca ustvarila glasbo na osnovi slike.

*»The only limit to AI is
human imagination.« (Chris Duffey)*

8 SEZNAM UPORABLJENIH VIROV

- [1] DNEVNIK. *Humanoidni robot* (online). (Citirano 15 Januar 2022). Dostopno na naslovu: https://www.dnevnik.si/i/ag/2012/09/26/1042553995_1.jpg.
- [2] EVROPSKI PARLAMENT. *Primeri uporabe umetne inteligence v vsakdanjem življenju* (online). (Citirano 15 Januar 2022). Dostopno na naslovu: https://www.europarl.europa.eu/resources/library/images/20201019PHT89631/20201019PHT89631_original.jpg.
- [3] INŠTITUT FRANA RAMOVŠA. *ZRC - SAZU* (online). (Citirano 7 Januar 2022). Dostopno na naslovu: <https://isjfr.zrc-sazu.si/sl/terminologisce/slovarji/avtomatika/iskalnik?iztocnica=um%C3%A9na%20intelig%C3%A9na>.
- [4] J. CHEN. *Zgradba nevronske mreže* (online). (Citirano 7 Januar 2022). Dostopno na naslovu: [https://www.investopedia.com/thmb/dmfmF8iByVZy1daIOwzHggagidk=/660x0/filters:no_upscale\(\):max_bytes\(150000\):strip_icc\(\):format\(webp\)/dotdash_Final_Neural_Network_Apr_2020-01-5f4088dfda4c49d99a4d927c9a3a5ba0.jpg](https://www.investopedia.com/thmb/dmfmF8iByVZy1daIOwzHggagidk=/660x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/dotdash_Final_Neural_Network_Apr_2020-01-5f4088dfda4c49d99a4d927c9a3a5ba0.jpg).
- [5] K. KRZYK. *Delitev strojnega učenja* (online). (Citirano 7 Januar 2022). Dostopno na naslovu: https://miro.medium.com/max/1400/1*8wU0hfUY3UK_D8Y7tbIyFQ.png.
- [6] OPENAI. *Generiranje zvoka* (online). (Citirano 6 Februar 2022). Dostopno na naslovu: <https://openai.com/blog/jukebox/>.
- [7] OPENAI. *MuseNet* (online). (Citirano 2 Februar 2022). Dostopno na naslovu: <https://openai.com/blog/musenet/>.
- [8] OPENAI. *Jukebox* (online). (Citirano 6 Februar 2022). Dostopno na naslovu: <https://openai.com/blog/jukebox/>.
- [9] OPENAI. *Razvrščanje izvajalcev po žanrih* (online). (Citirano 2 Februar 2022). Dostopno na naslovu: <https://openai.com/blog/jukebox/>.

- [10] OPENAI. *Stiskanje zvoka* (online). (Citirano 6 Februar 2022). Dostopno na naslovu: <https://openai.com/blog/jukebox/>.
- [11] S. JOSEPH. *Področja uporabe umetne inteligence* (online). (Citirano 15 Januar 2022). Dostopno na naslovu: <https://i.pinimg.com/564x/e2/b5/1e/e2b51ea857f78382753dcc9812a04645.jpg>.
- [12] S. BEHNKE. *Primer globokega učenja* (online). (Citirano 15 Januar 2022). Dostopno na naslovu: https://upload.wikimedia.org/wikipedia/commons/thumb/2/26/Deep_Learning.jpg/300px-Deep_Learning.jpg.
- [13] UNIVERZA V HELSINKIH. *Elementi AI-ja* (online). (Citirano 15 Januar 2022). Dostopno na naslovu: <https://course.elementsofai.com/sl/1/2>.
- [14] UNIVERZA V HELSINKIH. *Elementi AI-ja* (online). (Citirano 7 Januar 2022). Dostopno na naslovu: <https://course.elementsofai.com/sl/1/1>.
- [15] WIKIPEDIA. *Data science* (online). (Citirano 15 Januar 2022). Dostopno na naslovu: https://en.wikipedia.org/wiki/Data_science.
- [16] WIKIPEDIA. *Deep learning* (online). (Citirano 15 Januar 2022). Dostopno na naslovu: https://en.wikipedia.org/wiki/Deep_learning.
- [17] WIKIPEDIA. *Globoko učenje* (online). (Citirano 15 Januar 2022). Dostopno na naslovu: https://sl.wikipedia.org/wiki/Globoko_u%C4%8Denje.
- [18] WIKIPEDIA. *Neural network* (online). (Citirano 20 Januar 2022). Dostopno na naslovu: https://en.wikipedia.org/wiki/Neural_network.
- [19] WIKIPEDIA. *Robotics* (online). (Citirano 16 Januar 2022). Dostopno na naslovu: <https://en.wikipedia.org/wiki/Robotics>.
- [20] WIKIPEDIA. *Strojno učenje* (online). (Citirano 7 Januar 2022). Dostopno na naslovu: https://sl.wikipedia.org/wiki/Strojno_u%C4%8Denje.

9 PRILOGE

Priloga 1 - Jukebox audio compressing Original.mp3

Priloga 2 - Jukebox audio compressing 8x.mp3

Priloga 3 - Jukebox audio compressing 32x.mp3

Priloga 4 - Jukebox audio compressing 128x.mp3

Priloga 5 - AIVA The story - Soft Piano Solo - 4 4 - G major.mp3

Priloga 6 - AIVA Composition 1 - Piano Solo - 4 4 - C major.mp3

Priloga 7 - AIVA Composition 2 - Pop Ensemble - 4 4 - Bb minor.mp3

Priloga 8 - Jukebox Sample.wav

Priloga 9 - MuseNet improvises Video Games 1.mp3

Priloga 10 - MuseNet improvises Video Games from Beethoven's Für Elise.mp3

Priloga 11 - MuseNet improvises Video Games 2.wav

Priloga 12 - Magenta BachPerformance.mp3

Priloga 13 - Magenta BachPerformance 1.mp3

Priloga 14 - Magenta BachPerformance 2.mp3

Priloga 15 - BachPerformance_rnn.mag (»Bundle file«)