

# **Lumen - Data Science 2023 Dokumentacija**

# **Sadržaj**

Sadržaj	2
1. Uvod	4
2. Pregled	5
3. Razumijevanje podataka	6
4. Formulacija problema	15
5. Podjela podataka	16
6. Generacija značajki	17
7. Duboko učenje	19
7. Interpretacija modela	22
8. Treniranje	24



# 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.

Glazbeni instrumenti mogu se klasificirati na više načina, na temelju njihove konstrukcije, proizvodnje zvuka ili kulturnog podrijetla. Ova dokumentacija pružit će sveobuhvatan pregled najčešćih sustava klasifikacije koji se koriste u glazbi.

## 2. Pregled

Kratki pregled rada:

1. **Razumijevanje podataka:** U ovom odjeljku ćemo govoriti u augmentaciji audio podataka u svrhu generiranja veće količine podataka potrebnih za duboko učenje.
2. **Formulacija problema:** Nakon augmentacije audiozapisa predstaviti ćemo glavne ideje našeg pristupa.
3. **Konstrukcija modela:** Ovo poglavlje govori o svim komponentama našeg modela.
4. **Rezultati, analiza i zaključak:** Izlažemo i pokušavamo interpretirati dobivene podatke.

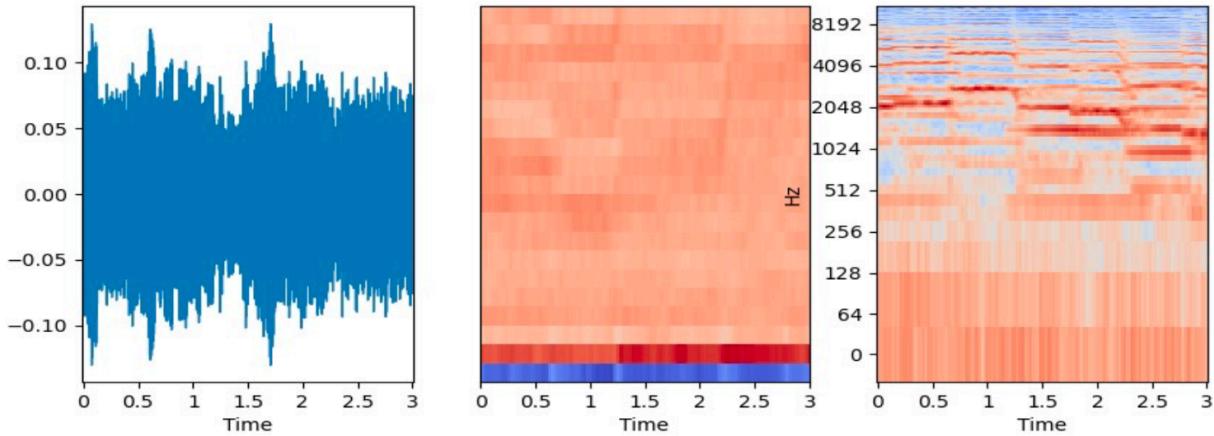
### 3. Razumijevanje podataka

#### 3.1 PREGLED PODATAKA

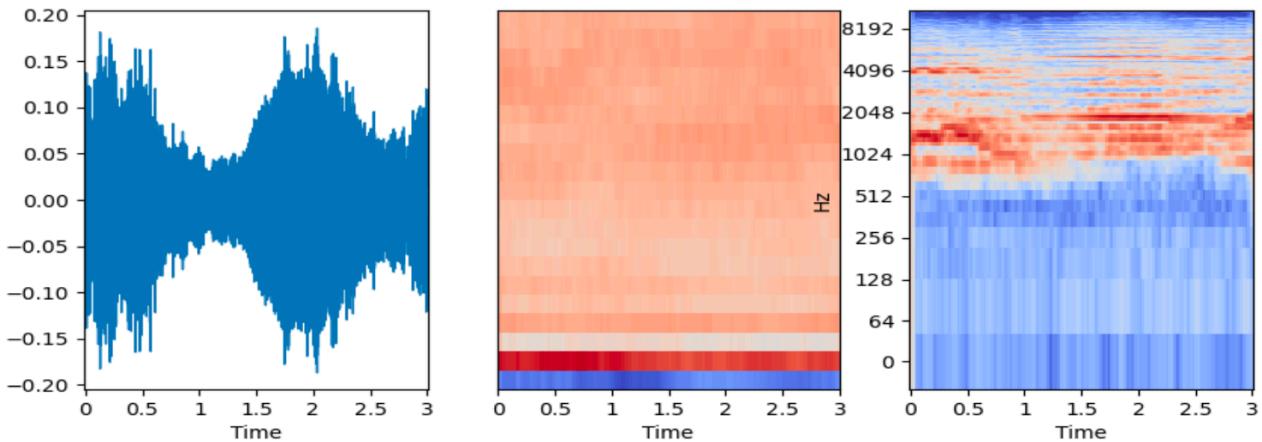
Dobivena baza podataka za rješavanje nagradnog zadatka sastoji se od 6705 audio datoteka u 16-bitnom stereo wav formatu uzorkovanih na 44,1 kHz. S obzirom da nam je za učenje modela potrebno puno veća baza podataka, postojeće podatke smo iskoristili da kreiranjem novih promijenjenih podataka na sljedeće načine:

- Kombiniranje i spajanje dvaju ili više različitih audio datoteka u jednu.
- Nasumično nadodavanje šuma audio podacima.
- Izmjena već dobivenih audio podataka (npr. skaliranje, prodljivanje, pomicanje itd.)

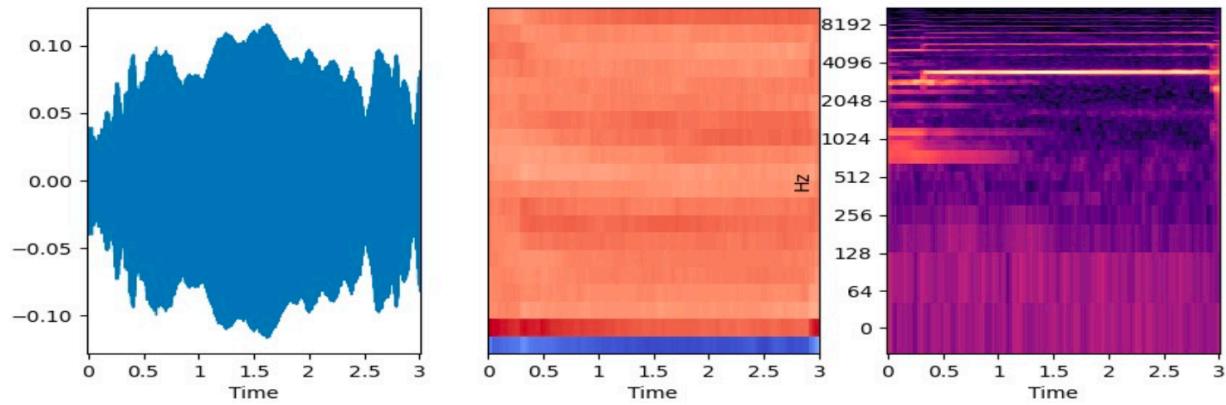
Nakon provedenih promjena, naša baza audio podataka sada sadrži preko 900 000 audio datoteka što nam je i više nego dovoljno za učenje modela.



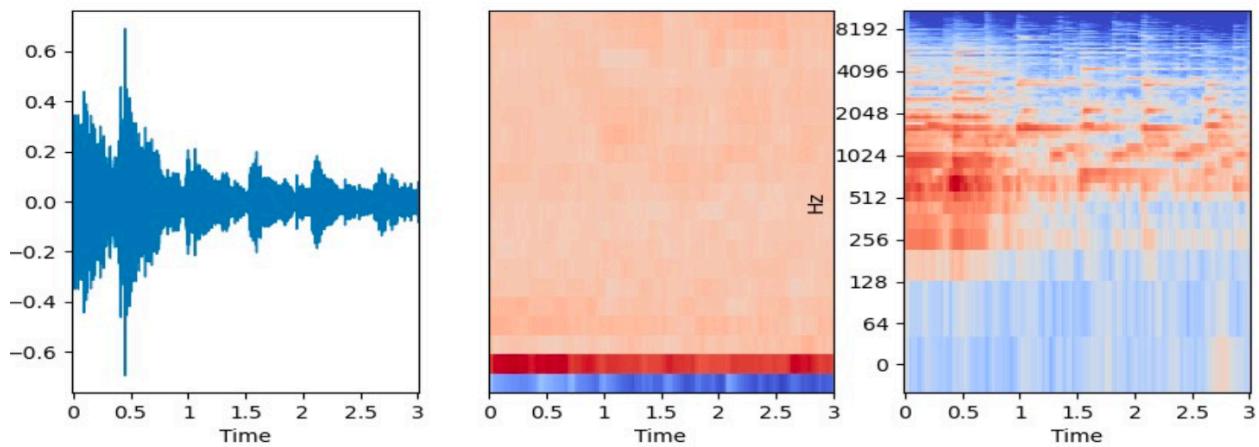
Slika 3.1 Valni izgled, prikaz mfcc-a i melspektograma za Cel kategoriju instrumenata



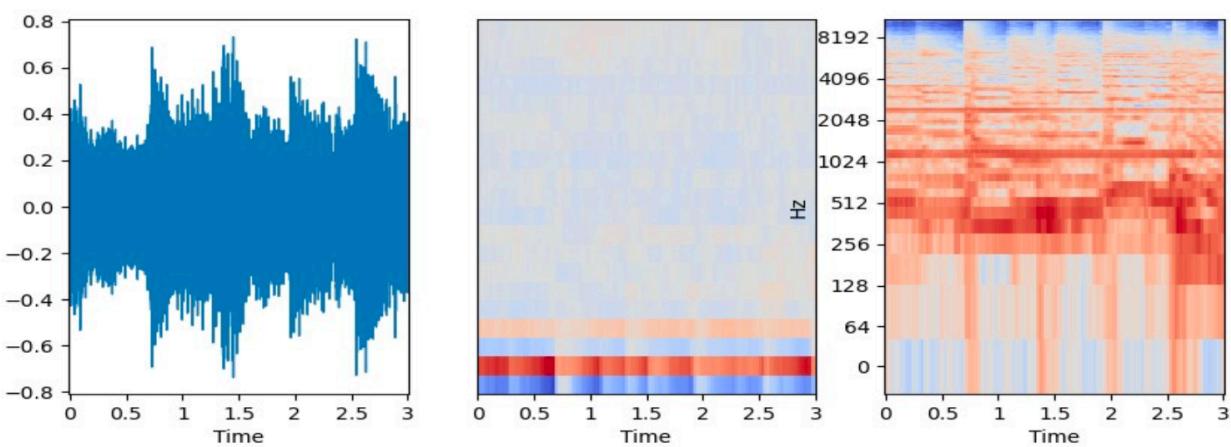
Slika 3.2 Valni izgled, prikaz mfcc-a i melspektograma za Cla kategoriju instrumenata



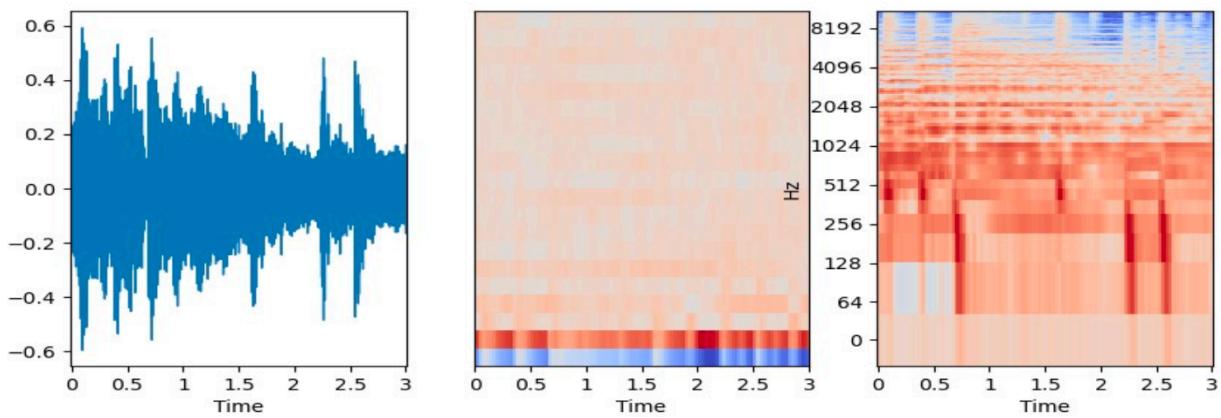
**Slika 3.3** Valni izgled, prikaz mfcc-a i melspektograma za Flu kategoriju instrumenata



**Slika 3.4** Valni izgled, prikaz mfcc-a i melspektograma za Gac kategoriju instrumenata



**Slika 3.5** Valni izgled, prikaz mfcc-a i melspektograma za Gel kategoriju instrumenata



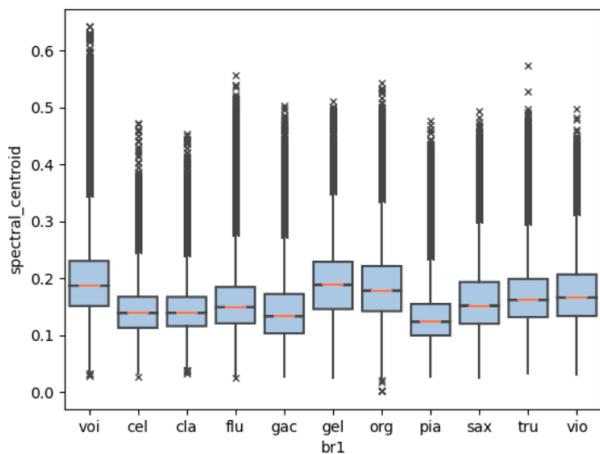
**Slika 3.6** Valni izgled, prikaz mfcc-a i melspektograma za Org kategoriju instrumenata

## 2.2 VIZUALIZACIJA I EKSTRAKCIJA ZNAČAJKI

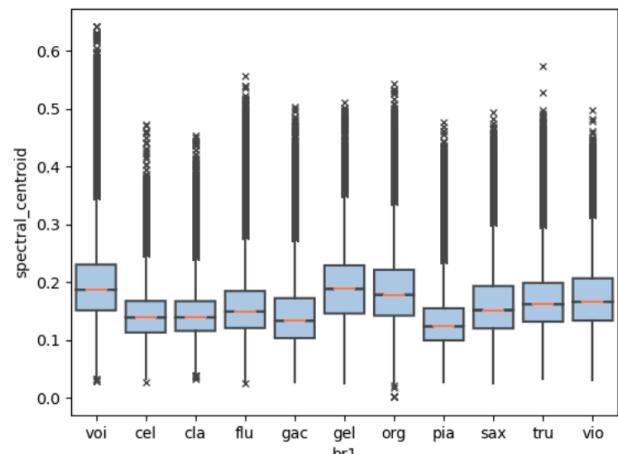
S obzirom da se do povoljnog rješenja problema nagradnog zadatka dolazi baratanjem audio datotekama čija vizualizacija i interpretacija nije toliko intuitivna te kategorička interferencija nije povoljna koristeći same auditivne podatke, prvi zadatak nam je bio evaluirati moguće značajke i ekstrahirati one koje su nam se činile kao najinformativnije.

Moguće značajke koje smo pretražili i ispitali su sljedeće:

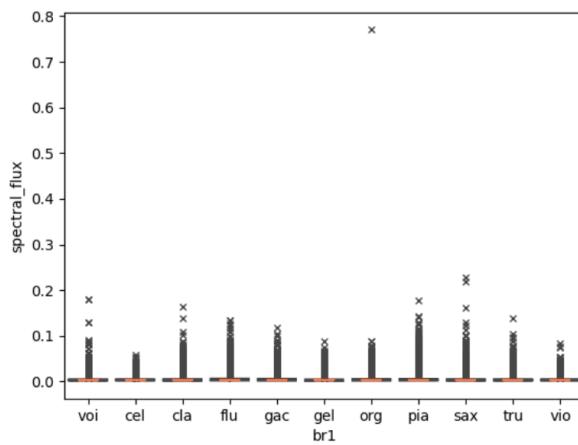
- Spektralni centroid
- Sprektralno raspršenje
- Sprektralna entropija
- Spektralni tok
- Spektralno opadanje



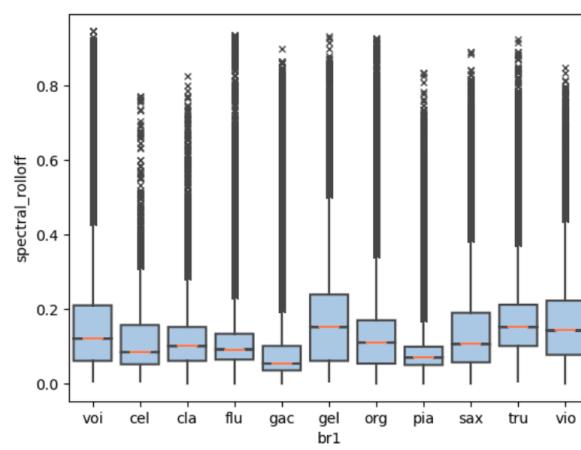
**Slika 3.7** Boxplot spektralnih centroida različitih instrumenata



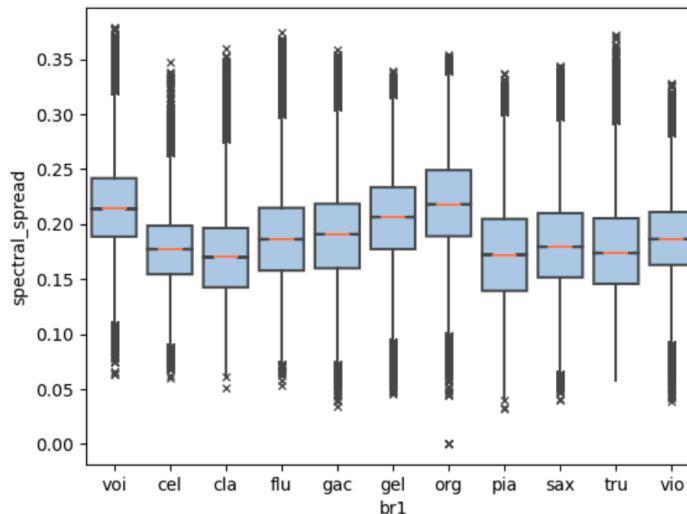
**Slika 3.8** Boxplot spektralne entropije različitih intrumenata



**Slika 3.9** Boxplot centralnog toka za različite instrumente



**Slika 3.10** Boxplot centralnog opadanja za različite instrumente



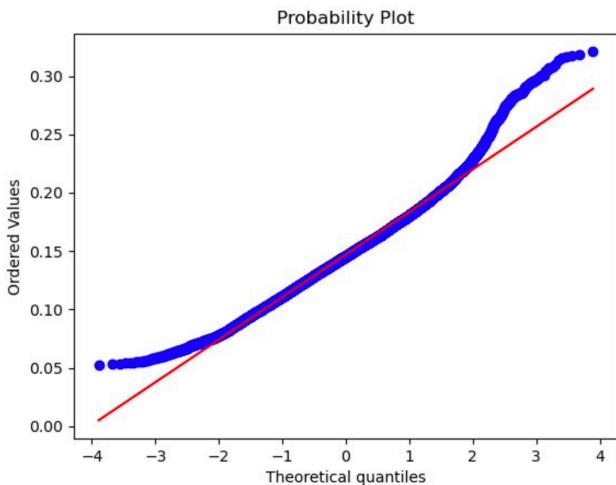
**Slika 3.11** Boxplot spektralnog raspršenja pojedinih instrumenata

Gledajući samo ove boxplotove, možemo uočiti neke minimalne razlike u distribucijama pojedinih značajki za pojedine instrumente, no ne možemo donijeti nikakve zaključke o utjecaju svake značajke na donošenje odluke o kategoriji instrumenta.

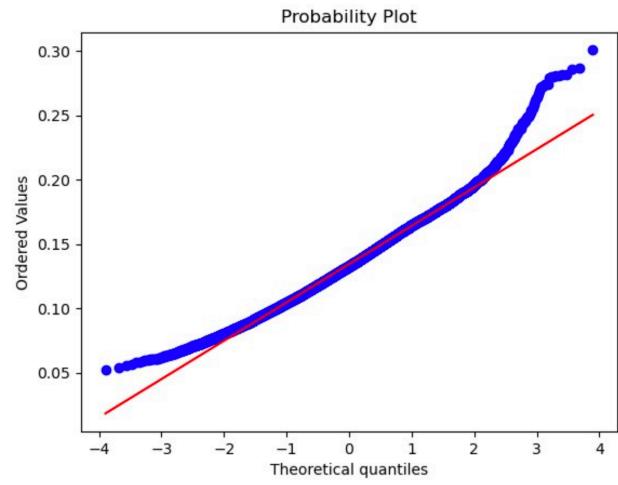
## 2.2.1 STATISTIČKO TESTIRANJE NAD PODACIMA

Da bismo mogli dobiti neki uvid u utjecajnost pojedinih značajki, provest ćemo par statističkih testova prije treniranja modela o kojem će se pričati u sljedećim poglavljima.

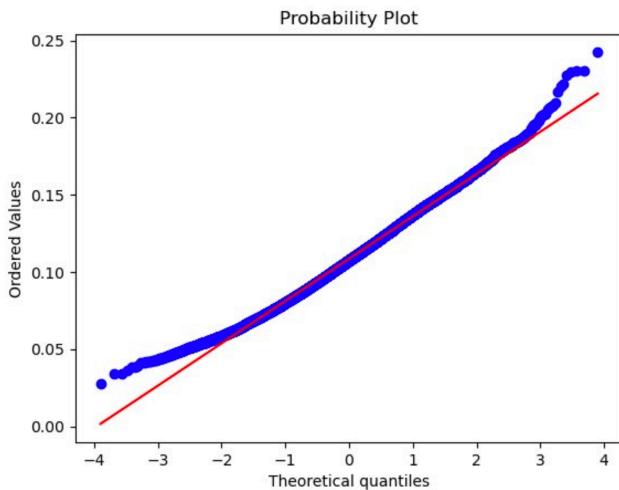
Testove koje želimo provesti nad podacima su: ANOVE/Kruskal-Wallisov test za kategoričke podatke, Tukey-Kramer test za parove kategoričkih podataka, međusobni T-testovi. Prije ispitivanja značajnosti razlika u distribucijama značajki za pojedine instrumente, ispitat ćemo normalnost samih distribucija s obzirom da je to glavni uvjet za provođenje određenih parametarskih testova. S obzirom da za svaki instrument imamo 4 različite značajke čije distribucije uspoređujemo, ukupan broj distribucija čiju normalnost moramo ispitati je 44, tako da nećemo prikazati svaki dobiveni rezultat testa nego ćemo prikazati par dobivenih vjerojatnosnih „QQ“ plot.



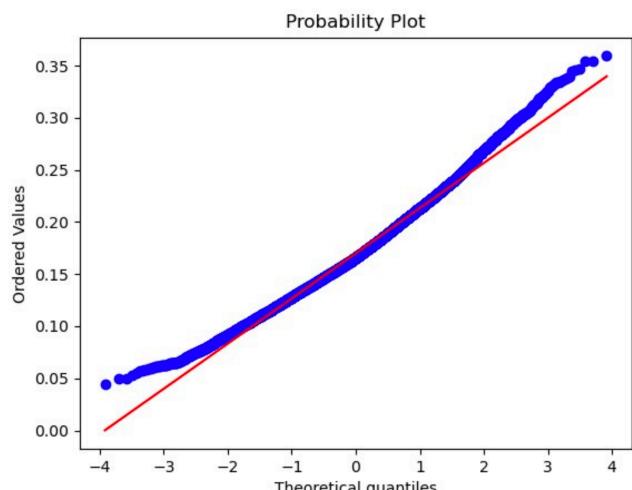
**Slika 3.12** QQ plot za spektralni centroid Cel instrumenata



**Slika 3.13** QQ plot za spektralni centroid Cla instrumenata



**Slika 3.14** QQ plot za spektralni centroid Vio instrumenata



**Slika 3.15** QQ plot za spektralni centroid Pia instrumenata

Kao što je i uočljivo preko grafova, prikazane distribucije teže normalnoj, a primjenom Lillieforsove inačice Kolomogorov-Smirnovog testa normalnosti dobili smo iste zaključke i za preostale distribucije. Dakle, sljedeće parametarske testove možemo s visokom sigurnošću provesti pod pretpostavkom normalnosti distribucije.

Prvotni statistički test koji smo proveli nad podacima jest jednofaktorska ANOVA. Jednofaktorska ANOVA se primjenjuje kada želimo provjeriti jednakost sredina različitih populacija unutar iste baze podataka, tj. u ovom slučaju ispitujemo jednakost značajki među svim instrumentima zajedno. Hipoteze koje ispitujemo provođenjem ANOVE su sljedeće:

$$H_0: \mu_1 = \mu_2 = \dots = \mu_n$$

$$H_1: \text{barem dvije sredine nisu jednake}$$

Provodenjem testova dobivamo sljedeće ispise:

	sum_sq	df	F	PR(>F)
C(br1)	444.739384	10.0	12716.239249	0.0
Residual	3119.321528	891894.0	NaN	NaN

Slika 3.16 ANOVA ispis s obzirom na spektralni centroid

	sum_sq	df	F	PR(>F)
C(br1)	30012.188783	10.0	11386.292202	0.0
Residual	235086.985542	891894.0	NaN	NaN

Slika 3.17 ANOVA ispis s obzirom na spektralnu entropiju

	sum_sq	df	F	PR(>F)
C(br1)	772.885885	10.0	6941.517989	0.0
Residual	9930.569723	891894.0	NaN	NaN

Slika 3.18 ANOVA ispis s obzirom na spektralnog opadanja

	sum_sq	df	F	PR(>F)
C(br1)	0.138764	10.0	781.820835	0.0
Residual	15.830084	891894.0	NaN	NaN

Slika 3.19 ANOVA ispis s obzirom na spektralno raspršenje

Po dobivenim rezultatima možemo vidimo da u p-vrijednosti značajno malene, tj. uopće ih nema. Kako za ANOVA-u vrijedi sljedeće: „Ako je p-vrijednost manja od razine značajnosti  $\alpha$ , odbacujemo hipotezu  $H_0$  u korist hipoteze  $H_1$ ”, možemo zaključiti da na određenim razinama sigurnosti postoji značajna razlika barem između dvije sredine svakih distribucija značajki za pojedine instrumente. Ovime smo osporili naše intuitivne zaključke na koje nas je vizualni prikaz podataka naveo.

S ovim saznanje želimo otkriti između koji instrumenata su najizražajnije te razlike, time određujući koja dva instrumenta možemo s lakoćom razlikovati i po kojim značajkama. Do toga ćemo doći provodenjem Tukey-Kramer testa za parove kategoričkih podataka. Za Tukey-Kramerov test ispituju se sljedeće hipoteze:

$$H_0: \mu_i = \mu_j$$

$$H_1: \mu_i \neq \mu_j$$

gdje  $i$  i  $j$  predstavljaju klasu instrumenata.

Ispisi dobiveni provođenjem testova su sljedeći:

	group1	group2	Diff	Lower	Upper	q-value	p-value
0	voi	cel	0.051926	0.050212	0.053641	137.869880	0.001000
1	voi	cla	0.064555	0.062853	0.066256	172.692097	0.001000
2	voi	flu	0.030536	0.028195	0.032877	59.377296	0.001000
3	voi	gac	0.034251	0.032360	0.036142	82.451777	0.001000
4	voi	gel	0.003990	0.002964	0.005016	17.702791	0.001000
5	voi	org	0.003440	-0.000356	0.007236	4.124885	0.118073
6	voi	pia	0.075893	0.074499	0.077287	247.824979	0.001000
7	voi	sax	0.050159	0.049075	0.051962	159.329098	0.001000
8	voi	tru	0.031572	0.028501	0.034643	46.792555	0.001000
9	voi	vio	0.021516	0.020069	0.022963	67.692427	0.001000
10	cel	cla	0.012629	0.010345	0.014912	25.177252	0.001000
11	cel	flu	0.021390	0.018598	0.024183	34.868075	0.001000
12	cel	gac	0.017675	0.015247	0.020103	33.142396	0.001000
13	cel	gel	0.047936	0.046100	0.049772	118.854557	0.001000
14	cel	org	0.048486	0.044397	0.052576	53.965649	0.001000
15	cel	pia	0.023967	0.021903	0.026031	52.852284	0.001000
16	cel	sax	0.001408	-0.000690	0.003505	3.054424	0.528997
17	cel	tru	0.020354	0.016926	0.023782	27.028359	0.001000
18	cel	vio	0.030411	0.028310	0.032511	65.910796	0.001000
19	cla	flu	0.034019	0.031234	0.036803	55.610497	0.001000
20	cla	gac	0.030304	0.027885	0.032722	57.034583	0.001000
21	cla	gel	0.060564	0.058741	0.062388	151.152151	0.001000
22	cla	org	0.061115	0.057031	0.065200	68.110634	0.001000
23	cla	pia	0.011338	0.009285	0.013392	25.132950	0.001000
24	cla	sax	0.014036	0.011949	0.016124	30.609734	0.001000
25	cla	tru	0.032983	0.029561	0.036404	43.879873	0.001000
26	cla	vio	0.043839	0.040949	0.045129	93.748409	0.001000
27	flu	gac	0.003715	0.000811	0.006619	5.822864	0.001000
28	flu	gel	0.026546	0.024114	0.028977	49.699678	0.001000
29	flu	org	0.027896	0.022707	0.031486	28.099569	0.001000
30	flu	pia	0.045357	0.042749	0.047965	79.165467	0.001000
31	flu	sax	0.019982	0.017348	0.022617	34.523752	0.001000
32	flu	tru	0.001836	-0.002744	0.004816	1.247407	0.900000
33	flu	vio	0.009920	0.006384	0.011657	15.573188	0.001000
34	gac	gel	0.030261	0.028259	0.032762	68.814410	0.001000
35	gac	org	0.030811	0.026644	0.034978	33.658863	0.001000
36	gac	pia	0.041642	0.039429	0.043855	85.655582	0.001000
37	gac	sax	0.016267	0.014023	0.018512	32.993432	0.001000
38	gac	tru	0.002679	-0.000841	0.006199	3.464885	0.334870
39	gac	vio	0.012735	0.010489	0.014982	25.803663	0.001000
40	gel	org	0.000551	-0.003302	0.004403	0.650611	0.900000
41	gel	pia	0.071903	0.070362	0.073444	212.406393	0.001000
42	gel	sax	0.046528	0.044943	0.048114	133.568710	0.001000
43	gel	tru	0.027582	0.024441	0.030722	39.975005	0.001000
44	gel	vio	0.017525	0.015936	0.019114	50.208474	0.001000
45	org	pia	0.072453	0.068487	0.076420	83.154860	0.001000
46	org	sax	0.047079	0.043095	0.051063	53.793690	0.001000
47	org	tru	0.028132	0.022314	0.032951	26.574873	0.001000
48	org	vio	0.018076	0.014091	0.022061	20.647505	0.001000
49	pia	sax	0.025375	0.023529	0.027220	62.597526	0.001000
50	pia	tru	0.044321	0.041042	0.047601	61.519261	0.001000
51	pia	vio	0.054377	0.052529	0.056225	133.945545	0.001000
52	sax	tru	0.018946	0.015646	0.022247	26.128973	0.001000
53	sax	vio	0.029803	0.027117	0.030888	70.021341	0.001000
54	tru	vio	0.010056	0.006754	0.013359	13.862231	0.001000

Slika 3.20 Ispisi Tukey-Kramer testa s obzirom na spektralni centroid

Slika 3.21 Ispisi Tukey-Kramer testa s obzurom na spektralnu entropiju

	group1	group2	Diff	Lower	Upper	q-value	p-value		group1	group2	Diff	Lower	Upper	q-value	p-value
0	voi	cel	0.000334	0.000223	0.000445	13.676905	0.001000	0	voi	cel	0.047243	0.043923	0.050562	64.783811	0.001000
1	voi	cla	0.000182	0.000071	0.000292	7.494737	0.001000	1	voi	cla	0.061490	0.058196	0.064785	84.956956	0.001000
2	voi	flu	0.001437	0.001285	0.001589	43.096841	0.001000	2	voi	flu	0.053651	0.049118	0.058183	53.880180	0.001000
3	voi	gac	0.000111	-0.000012	0.000233	4.183901	0.122791	3	voi	gac	0.046274	0.042613	0.049935	57.532074	0.001000
4	voi	gel	0.000216	0.000150	0.000283	14.783214	0.001000	4	voi	gel	0.003644	0.001658	0.005631	8.350494	0.001000
5	voi	org	0.002304	0.002058	0.002551	42.624137	0.001000	5	voi	org	0.057887	0.050538	0.065237	35.852355	0.001000
6	voi	pia	0.001881	0.000991	0.001172	54.458119	0.001000	6	voi	pia	0.086568	0.083861	0.089259	145.984095	0.001000
7	voi	sax	0.000443	0.000350	0.000537	21.572640	0.001000	7	voi	sax	0.039626	0.036832	0.042421	64.546638	0.001000
8	voi	tru	0.000026	-0.000173	0.000225	0.592440	0.900000	8	voi	tru	0.023370	0.017423	0.029317	17.888585	0.001000
9	voi	vio	0.000056	-0.000038	0.000150	2.727885	0.672611	9	voi	vio	0.013181	0.010379	0.015982	21.417730	0.001000
10	cel	cla	0.000152	0.000004	0.000360	4.684145	0.037440	10	cel	cla	0.014248	0.009827	0.018668	14.670379	0.001000
11	cel	flu	0.0001103	0.000922	0.0001284	27.731914	0.001000	11	cel	flu	0.006408	0.001001	0.011815	5.394731	0.006405
12	cel	gac	0.000045	0.000287	0.000602	12.855511	0.001000	12	cel	gac	0.000969	-0.003731	0.005669	0.938289	0.900000
13	cel	gel	0.000550	0.000431	0.000669	21.034208	0.001000	13	cel	gel	0.050887	0.047333	0.054442	65.156472	0.001000
14	cel	org	0.001971	0.001705	0.002236	33.827748	0.001000	14	cel	org	0.010645	0.008726	0.018563	6.118948	0.001000
15	cel	pia	0.000747	0.000613	0.000881	25.417152	0.001000	15	cel	pia	0.039317	0.035320	0.043313	44.779246	0.001000
16	cel	sax	0.000109	-0.000027	0.000246	3.664685	0.253038	16	cel	sax	0.007617	0.003555	0.011678	8.535876	0.001000
17	cel	tru	0.000360	0.000138	0.000582	7.371064	0.001000	17	cel	tru	0.023873	0.017236	0.030510	16.372799	0.001000
18	cel	vio	0.000278	0.000142	0.000414	9.285258	0.001000	18	cel	vio	0.060424	0.056357	0.064490	67.637453	0.001000
19	cla	flu	0.001255	0.001075	0.001436	31.650902	0.001000	19	cla	flu	0.007840	0.006248	0.013231	6.619014	0.001000
20	cla	gac	0.000292	0.000135	0.000449	8.481543	0.001000	20	cla	gac	0.015216	0.010534	0.019899	14.791210	0.001000
21	cla	gel	0.000398	0.000279	0.000516	15.308510	0.001000	21	cla	gel	0.065135	0.061603	0.068666	83.957167	0.001000
22	cla	org	0.002123	0.001858	0.002388	36.490588	0.001000	22	cla	org	0.003603	0.004305	0.011511	2.073819	0.900000
23	cla	pia	0.000900	0.000766	0.001033	30.756864	0.001000	23	cla	pia	0.025069	0.021093	0.029045	28.700076	0.001000
24	cla	sax	0.000262	0.000126	0.000397	8.886788	0.001000	24	cla	sax	0.021864	0.017823	0.025966	24.625791	0.001000
25	cla	tru	0.000208	-0.000014	0.000429	4.259073	0.000916	25	cla	tru	0.038121	0.031496	0.044745	26.193070	0.001000
26	cla	vio	0.000125	-0.000010	0.000261	4.213966	0.0009423	26	cla	vio	0.074671	0.070625	0.078717	84.004231	0.001000
27	flu	gac	0.001547	0.001359	0.001736	37.411224	0.001000	27	flu	gac	0.087377	0.001754	0.013000	5.971545	0.001217
28	flu	gel	0.001653	0.001495	0.001811	47.733876	0.001000	28	flu	gel	0.057295	0.052588	0.062003	55.401744	0.001000
29	flu	org	0.000868	0.000583	0.001152	13.876226	0.001000	29	flu	org	0.004237	-0.004262	0.012736	2.269261	0.874321
30	flu	pia	0.000356	0.000187	0.000525	9.575937	0.001000	30	flu	pia	0.032909	0.027859	0.037959	29.665635	0.001000
31	flu	sax	0.000993	0.000823	0.001164	26.474419	0.001000	31	flu	sax	0.014024	0.008923	0.019126	12.514191	0.001000
32	flu	tru	0.001463	0.001218	0.001708	27.168241	0.001000	32	flu	tru	0.030281	0.022961	0.037601	18.831075	0.001000
33	flu	vio	0.001381	0.001210	0.001526	36.767843	0.001000	33	flu	vio	0.066831	0.061726	0.071936	59.598768	0.001000
34	gac	gel	0.000106	-0.000024	0.000235	3.709036	0.239536	34	gac	gel	0.049918	0.046043	0.053794	58.628465	0.001000
35	gac	org	0.002415	0.002145	0.002685	40.691620	0.001000	35	gac	org	0.011614	0.0083546	0.019682	6.552412	0.001000
36	gac	pia	0.001192	0.001048	0.001335	37.810695	0.001000	36	gac	pia	0.040286	0.036001	0.044570	42.779572	0.001000
37	gac	sax	0.000554	0.000408	0.000700	17.330523	0.001000	37	gac	sax	0.006648	0.002302	0.010993	6.963593	0.001000
38	gac	tru	0.000085	-0.000144	0.000313	1.687893	0.900000	38	gac	tru	0.022904	0.016090	0.029719	15.299506	0.001000
39	gac	vio	0.000167	0.000021	0.000312	5.210893	0.010427	39	gac	vio	0.059455	0.055105	0.063805	62.216147	0.001000
40	gel	org	0.002521	0.002271	0.002770	45.937780	0.001000	40	gel	org	0.061532	0.054073	0.068991	37.551762	0.001000
41	gel	pia	0.001297	0.001197	0.001397	59.189134	0.001000	41	gel	pia	0.090204	0.087221	0.093188	137.624488	0.001000
42	gel	sax	0.000660	0.000557	0.000762	29.201688	0.001000	42	gel	sax	0.043271	0.040200	0.046341	64.154830	0.001000
43	gel	tru	0.0000190	-0.000014	0.000394	4.250191	0.092542	43	gel	tru	0.027014	0.020933	0.033095	20.221237	0.001000
44	gel	vio	0.000272	0.000169	0.000375	12.030607	0.001000	44	gel	vio	0.009536	0.006460	0.012613	14.110377	0.001000
45	org	pia	0.001223	0.000966	0.001480	21.653970	0.001000	45	org	pia	0.028672	0.020993	0.036351	16.995593	0.001000
46	org	sax	0.001861	0.001603	0.002119	32.798400	0.001000	46	org	sax	0.018261	0.010548	0.025975	10.776713	0.001000
47	org	tru	0.002330	0.002018	0.002643	33.954137	0.001000	47	org	tru	0.034518	0.025188	0.043848	16.840588	0.001000
48	org	vio	0.0002248	0.0001990	0.0002507	39.610632	0.001000	48	org	vio	0.071068	0.063352	0.078784	41.926642	0.001000
49	pia	sax	0.000638	0.000518	0.000757	24.267341	0.001000	49	pia	sax	0.046933	0.043361	0.050506	59.798228	0.001000
50	pia	tru	0.001107	0.000895	0.001320	23.783248	0.001000	50	pia	tru	0.063190	0.056840	0.069540	45.299750	0.001000
51	pia	vio	0.001825	0.000905	0.001145	38.944162	0.001000	51	pia	vio	0.099740	0.096162	0.103318	126.890227	0.001000
52	sax	tru	0.000469	0.000255	0.000683	9.984347	0.001000	52	sax	tru	0.016256	0.009866	0.022647	11.578848	0.001000
53	sax	vio	0.000387	0.000265	0.000509	14.420576	0.001000	53	sax	vio	0.052807	0.049156	0.056458	65.845827	0.001000
54	tru	vio	0.000082	-0.000132	0.000296	1.746190	0.900000	54	tru	vio	0.036551	0.030157	0.042944	26.021531	0.001000

**2.2.2 ZAKLJUČAK O PODACIMA**

Možemo uočiti da je postoje „statistički značajne“ razlike među određenim klasama instrumenata s obzirom da sve promatrane značajke naših audio datoteka. Ovakvi rezultati se čine pozitivnima i mogli bismo odmah pomisliti kako se cijeli ovaj zadatak može riješiti koristeći SAMO statističko testiranje i zaključivanje.

No, statističko zaključivanje samo po sebi ipak nije dovoljan alat za rješavanje problema kategorizacije. Naime, problem kategorizacije zahtijeva dublje razumijevanje značajki koje su prisutne u audio datotekama te njihovu interpretaciju u odnosu na ciljnu kategoriju. Statističko zaključivanje može dati informaciju o tome jesu li dvije skupine značajki slične ili različite, ali ne i o tome kako se one trebaju kategorizirati u skladu s ciljem zadatka. Kategorizacija zahtijeva stručno znanje i iskustvo u području koje se proučava te metode strojnog učenja koje se mogu koristiti za automatsko kategoriziranje podataka. Stoga je potrebno kombinirati statističko testiranje s drugim alatima i tehnikama kako bi se postigla pouzdana i precizna kategorizacija podataka.

Jedno, što sigurno možemo izvući kao zaključak jest: **klasificiranje audio datoteka moguće je uz pažljivo treniranje koristeći informacije značajki koje smo ispitali.**

**Slika 3.23 Ispis Tukey-Kramer testa s obzirom na spektralno opadanje**

## 2.3 OSTALE DOSTUPNE ZNAČAJKE

Osim već navedenih značajki, u našem radu smo se bavili i dodatnim dostupnim značajkama audio datotekama, a to su:

- Melspektrogram
- MFCC

**Mel Frequency Cepstral Coefficients** (MFCC) je često korištena značajka u obradi govornih signala i prepoznavanju govora. MFCC su linearno-kosinusni koeficijenti koji se računaju kao transformacija frekvencijskog spektra signala nakon primjene banks filtra koji aproksimira osjetljivost slušnog sustava na različite frekvencijske pojaseve. Ova transformacija se koristi za smanjenje dimenzionalnosti zvukova i eliminaciju nepotrebnih podataka te se obično primjenjuje u kombinaciji s drugim tehnikama obrade signala ili algoritama strojnog učenja kako bi se postigla bolja klasifikacija ili prepoznavanje govora.

**Melspektrogram** je prikaz spektrograma koji se dobiva primjenom Mel filtra na spektar audio signala. Mel filtri su nizovi trokutastih filtera koji se koriste za aproksimaciju osjetljivosti ljudskog uha na različite frekvencijske pojaseve. Kada se primijene na frekvencijski spektar signala, Mel filtri pretvaraju linearne razmaknute frekvencije u logaritamski razmaknute frekvencije koje su bliže onima koje percipira ljudsko uho. Kao rezultat primjene Mel filtra na signal, dobivamo spektrogram u kojem su horizontalne osi vremenska os i vertikalna os predstavlja Mel skalu frekvencije. Melspektrogrami se koriste za analizu zvuka u mnogim primjenama, uključujući prepoznavanje govora, obradu govornih signala i klasifikaciju zvukova.

Prikaz i jedne i druge značajke su vidljive na samome početku ovog poglavlja.

## 4. Formulacija problema

Za dati MFCC (Mel-frequency cepstrum) audiozapisa potrebno je predvidjeti koje instrumente sadrži koristeći spektar snage određenog instrumenta koji se temelji na linearnoj kosinusovoj transformaciji.

Na danu temu postoje već različita istraživanja, čitajući iste shvatili smo da bolje rezultate daju modeli temeljeni na dubokom učenju nego modeli temeljeni na strojnom učenju. Potaknuti tim znanjem odlučili smo se na duboko učenje s prilagođenom arhitekturom. Naš pristup radi u jednom koraku.

### Klasifikacija:

#### Prednosti:

**Vjerovatnosc boljih rezultata** - temeljeno na različitim člancima duboko učenje daje bolje rezultate nego strojno učenje u konkretno ovaj temi.

#### Mogući problemi:

**Kompleksnost** - potrebno je razlučiti između 11 instrumenata što podiže razinu potrebnih podataka i faktor neodlučivosti između određenih instrumenata.

**Predmet istraživanja** - dana tema i obrade signala su još teme različitih znanstvenih istraživanja te ne postoji još arhitekture koje s velikom preciznošću mogu klasificirati određene instrumente.

Pokazalo se da su u našem slučaju problemi bili veći od prednosti, iako pri treniranju na suho, te smo prešli na drugačiji pred treniran model koji koristi sličan pristup, uz neke preinake, model u pitanju je YAMNet.

YAMNet (Yet Another Merging Network) je duboko učenje model koji se koristi za klasifikaciju zvuka. YAMNet ekstrahira značajke iz audio signala koristeći mrežu konvolucijskih neurona.

Model se sastoji od dvije glavne komponente: prednji dio koji koristi konvolucijsku neuronsku mrežu za ekstrakciju značajki iz audio signala, te stražnji dio koji se sastoji od potpuno povezanog sloja koji vrši klasifikaciju zvuka.

Ekstrakcija značajki u YAMNet-u radi se na sljedeći način: najprije se audio signal uzorkuje na 16 kHz i smanjuje se u trajanju na 3 sekunde. Zatim se signal prolazi kroz banku filtara kako bi se dobili spektrogrami signala. Svaki spektrogram podijeljen je na 256 vremenskih segmenata, a zatim se svaki segment obrađuje konvolucijskom mrežom. Konvolucijska mreža se sastoji od 14 slojeva, a njezine značajke se koriste za izgradnju reprezentacije zvuka.

Nakon prolaska kroz konvolucijsku mrežu, dobivaju se značajke koje se zatim šalju u potpuno povezani sloj za klasifikaciju zvuka. Potpuno povezani sloj sadrži 521 klase zvuka, uključujući razne vrste glazbenih instrumenata, ljudski govor, zvukove životinja i ostale zvukove.

YAMNet se trenira na velikom skupu podataka, pod nazivom AudioSet, koji sadrži preko 2 milijuna audio zapisa različitih zvukova. Kada se jednom trenira, model se može koristiti za klasifikaciju zvukova u stvarnom vremenu ili na postojećim audio zapisima. YAMNet se često koristi za razne zadatke poput prepoznavanja govora, prepoznavanja zvukova u prirodi ili u industriji, te za detekciju neželjenih zvukova.

## 5. Podjela podataka

Isprva je plan bio generirati kartezijev produkt svih kombinacija količine augmentacije, tako bi svaki audio sample bio pretvoren u njih cca. 700.

Također bi kombinirali po više instrumenata u isti sample, kako bi model naučio raditi i s takvim ulazima.

Svaki zvukovni zapis se obrađuje na različite načine kako bi se dobio set različitih uzoraka (tj. "augmented" podataka) koji će se koristiti za trening modela. Postoje različite tehnike obrade zvuka koje se primjenjuju, poput povećavanja/umanjivanja glasnoće, pomaka u vremenu, mijenjanja tona, dodavanja šuma, dodavanja "blinking" dijelova itd.

Svaki od tih obradaka se zatim pretvara u spektrogram i zatim se dobiva set MFCC (Mel Frequency Cepstral Coefficients) značajki, koji se koriste kao ulazni podaci za neuronsku mrežu koja će prepoznati koji instrument svira.

Na kraju, svi uzorci se spremaju u SQLite bazu podataka za daljnju uporabu u treningu modela. Proces se nastavlja sve dok se ne generira dovoljno podataka za trening.

Ovaj pristup je trebao biti optimalan, no nije rezultirao konvergencijom dubokog modela.

Idući je bio generirati random augmentirane podatke, te njegove yammnet embedding te retrenirati izlazne layere.

Za to smo koristili oko 200gb podataka generiranih generatorom koji ih feeda u neuralnu mrežu.

Stvara se 10 foldova na random u procesu, te deseti koristi za validaciju.

## 6. Generacija značajki

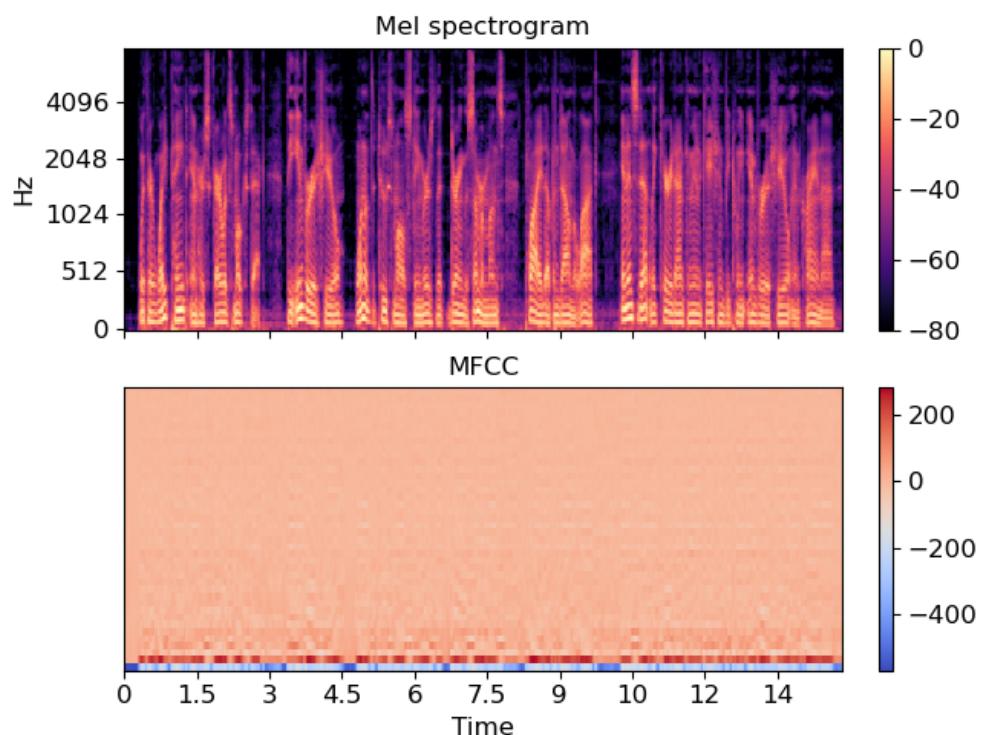
Prvi način koristio je MFCC.

MFCC (Mel-frequency cepstral coefficients) su koeficijenti koji se koriste za opisivanje zvukova u analizi zvuka i obradi govora. Oni se koriste za izvlačenje karakteristika zvuka iz vremenskog i frekvencijskog domena signala.

Mel-skala frekvencija je skala koja bolje odgovara načinu na koji čujemo zvukove nego linearna skala frekvencije. Ova skala razlikuje više niskih frekvencija nego visokih frekvencija, što odgovara našoj percepciji zvuka. MFCC koeficijenti se izračunavaju tako što se spektrogram zvuka (tj. vizualni prikaz spektra zvuka) podvrgne banki filtara koji su raspoređeni prema Mel-skali. Svaki filter se zatim uzima kao jedna od karakteristika signala, a za dobivanje konačnih MFCC koeficijenata se primjenjuje cepstralna analiza.

U glazbenoj industriji, MFCC koeficijenti se često koriste za prepoznavanje glazbenih instrumenata jer različiti instrumenti imaju karakteristične spekture i frekvencijske obrasce koji se mogu identificirati putem MFCC koeficijenata.

Od njih smo generirali ulaze veličine 130 sa 128 na ulazu, te si dokazali interpretabilnost rađenjem inverza pomoću librose, te slušajući par primjera kako bi znali da je razina detalja unutra dovoljna.



Slika 6.1 Prikaz MFCC spektrograma

## Drugi način

U finalnoj verziji koristili smo Yamnet Embedding-e kao ulaz.

YAMNet embedding predstavlja vektorski zapis zvuka koji se dobiva primjenom neuronske mreže YAMNet na audio snimak. YAMNet embedding se dobiva kao izlazni vektor neuronske mreže YAMNet, koja se trenira za klasifikaciju audiozapisa u više od 500 kategorija zvukova.

Iako YAMNet embedding nije konceptualno teško razumjeti, interpretacija samog vektorskog zapisa može biti teška zbog visoke dimenzionalnosti. Vektorski prostor u kojem su sadržani YAMNet embeddingovi ima velik broj dimenzija (dodijeljenih koeficijentima transformacije koju je naučila neuronska mreža), što može biti izazovno za interpretaciju i vizualizaciju.

Ipak, YAMNet embedding se može koristiti kao snažno sredstvo za klasifikaciju zvukova u nekoliko kategorija i za identifikaciju zvučnih obrazaca u audio snimcima. Osim toga, moguće je primijeniti tehniku smanjenja dimenzionalnosti na YAMNet embedding kako bi se omogućila vizualizacija vektorskog prostora i interpretacija pojedinih dimenzija.

Dimenzije izlaza su 1024, no mi koristimo 3 sekunde kao veličinu ulaza, tako da zapravo generiramo 6 ulaza u mfcc, flattenamo ih te prosljedimo našem custom završnom layeru.

Jedan od glavnih razloga zašto se koristi transfer learning je smanjenje vremena potrebnog za treniranje modela. Treniranje dubokih neuronskih mreža može trajati jako dugo, čak i na moćnim računalima. Ovo je posebno izazovno ako raspolažemo malom količinom podataka, a naš zadatak zahtijeva složene modele s velikim brojem parametara.

Transfer learning rješava ovaj problem tako što nam omogućuje korištenje prethodno treniranog modela koji je već naučio važne značajke. To znači da ne moramo u potpunosti trenirati novi model već samo prilagodimo dio koji je potreban našem zadatku, što može uvelike smanjiti vrijeme potrebno za treniranje. Osim toga, transfer learning često rezultira boljim performansama jer je prethodni model naučio generalne značajke, koje su često korisne i za naš specifičan zadatak.

Primjerice, u prethodnom primjeru korišten je prethodno trenirani model YAMNet za prepoznavanje zvukova u video zapisima, a zatim su se te naučene značajke upotrijebile u novom modelu za prepoznavanje glazbenih instrumenata. Korištenje naučenih značajki za novi zadatak također može smanjiti rizik od prenaučenosti, jer ne moramo trenirati model s velikim brojem parametara koji bi se mogao prilagoditi samo maloj količini podataka kojom raspolažemo.

Kod korišten u finalnoj verziji je upravo ovo:

```
def load_wav_16k_mono_augmented(filename):
    wav, sr = librosa.load(filename, sr=16000)

    rand_scale = random.uniform(0.7, 1.2)
    rand_shift = random.uniform(-0.2, 0.2)
    rand_stretch = random.uniform(0.8, 1.2)
    rand_pitch = random.uniform(-2, 2)
    rand_noise = random.uniform(0, 0.05)

    wav = scale_volume(wav, rand_scale)
    wav = shift_audio(wav, sr, rand_shift)
    wav = stretch_audio(wav, rand_stretch)
    wav = pitch_shift(wav, sr, rand_pitch)
    wav = add_noise(wav, rand_noise)

    return wav
```

Slika 6.2 Isječak koda za augmentiranje podataka

## 7. Duboko učenje

### Pokušaj VGG19.

Konvolucijske neuronske mreže su posebno pogodne za obradu slika i zvukova, što čini ovaj model idealnim za obradu ulaznih zvučnih signala koji se koriste za prepoznavanje glazbenih instrumenata.

Model koristi tri sloja konvolucijskih neuronskih mreža s različitim brojem filtera za ekstrakciju različitih značajki iz ulaznih zvučnih signala. Svaki sloj koristi aktivacijsku funkciju ReLU, koja je dokazano učinkovita u neuronskim mrežama za obradu slika i zvukova.

Nakon tri konvolucijska sloja slijede dva gusto povezana sloja, uključujući padajući sloj, koji pomaže u smanjenju overfittinga, tj. prenaučenosti modela, što ga čini manje osjetljivim na buku u ulaznim zvukovima.

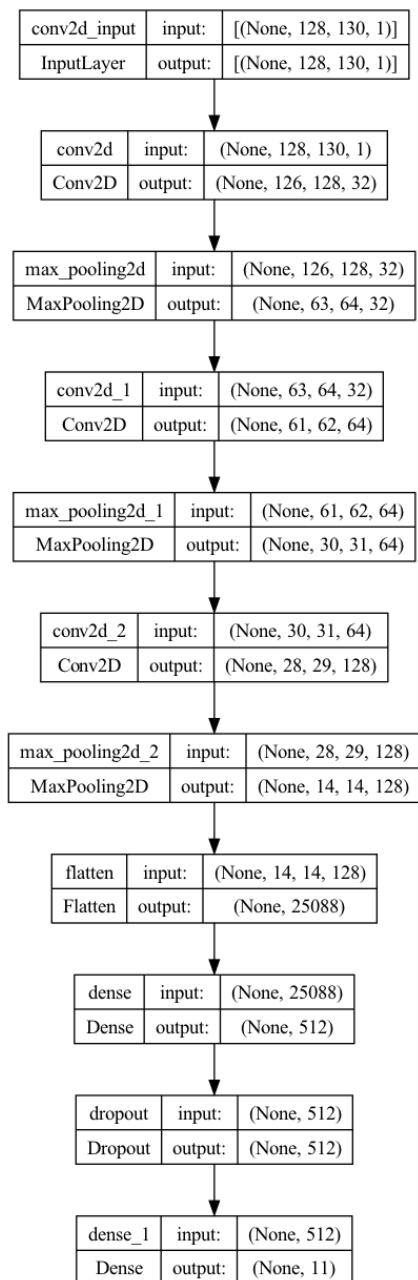
Konačno, izlazni sloj koristi funkciju aktivacije softmax za klasifikaciju ulaznih zvučnih signala u 11 različitih kategorija glazbenih instrumenata.

Svi ovi elementi čine ovaj model vrlo učinkovitim za prepoznavanje glazbenih instrumenata iz zvučnih signala.

### Zaključak

Model je slabo konvergirao vjerojatno jer je pogodniji ulaz sam spektrogram, a manje mfcc koeficijenti, makar jesu sažetiji prikaz.

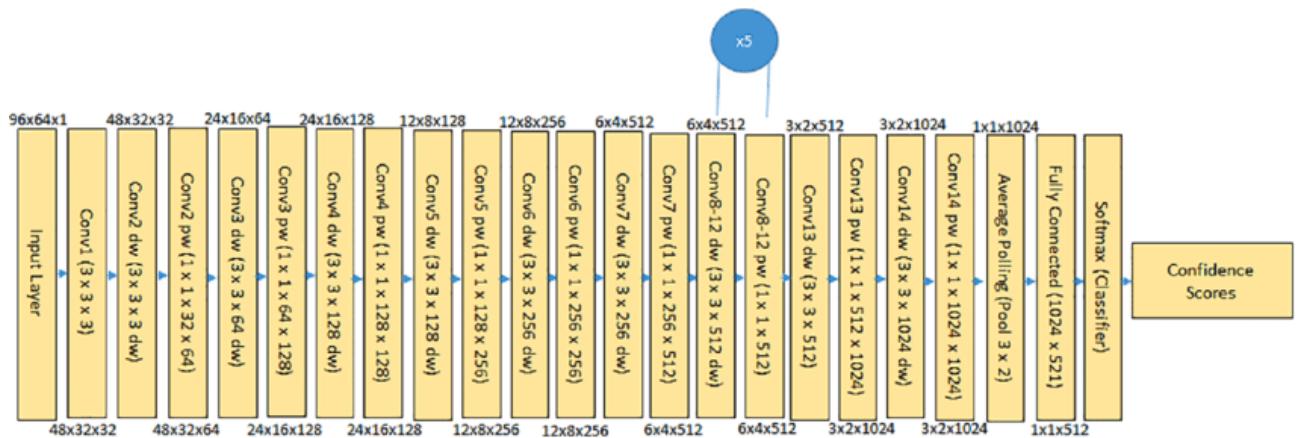
Potrebno bi bilo još mnogo treniranja.



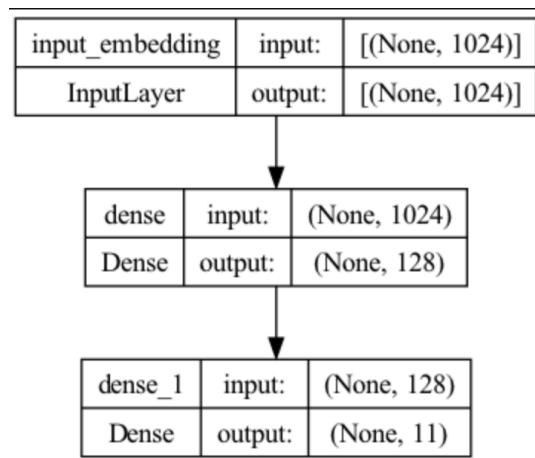
**Slika 6.3** Vizualizacija inicijalnog modela

## Drugi pokušaj:

Yamnnet - transfer learning



Slika 6.4 Prikaz yamnnet modela



Slika 6.5 Model koji koristimo u finalnoj inačici

**YAMNet** (Yet Another Music Network) je duboka neuronska mreža za prepoznavanje zvukova i glazbenih instrumenata koja je razvijena od strane Google AI-a. Mreža je trenirana na velikom skupu audiozapisa koji sadrže razne zvukove, uključujući i zvukove različitih glazbenih instrumenata.

YAMNet je pogodan za transfer learning u kontekstu prepoznavanja glazbenih instrumenata jer je treniran na velikom skupu podataka koji sadrže velik broj različitih zvukova. To omogućuje da mreža nauči generalne značajke zvukova i instrumenata, koje se mogu prenijeti na nove zadatke prepoznavanja instrumenata. Osim toga, YAMNet je dizajniran da se može koristiti kao osnova za transfer learning, što znači da se mreža može prethodno trenirati na velikom skupu zvukova, a zatim se fino podešava za specifičan problem prepoznavanja instrumenata, s manjim skupom podataka.

S obzirom na to da prepoznavanje glazbenih instrumenata zahtijeva veliku količinu podataka i specifične vještine prepoznavanja zvuka, YAMNet se pokazao vrlo korisnim u kontekstu transfer learninga, pružajući visoku preciznost prepoznavanja različitih glazbenih instrumenata.

Ovaj pristup se pokazao optimalnim, vrlo brzo stvarajući velike preciznosti, čak i oko 90% jer je dosta podataka redundantno (u drugom pristupu se generira 6 sampleova od svakog iz dataseta)

U stvarnom svijetu to je naravno manje.

Dobili smo Hamming Score 85% nad validacijskim podacima IRMAS-a.

Json validacije je na linku: <https://pastebin.com/raw/axNYm8A3>

## 7. Interpretacija modela

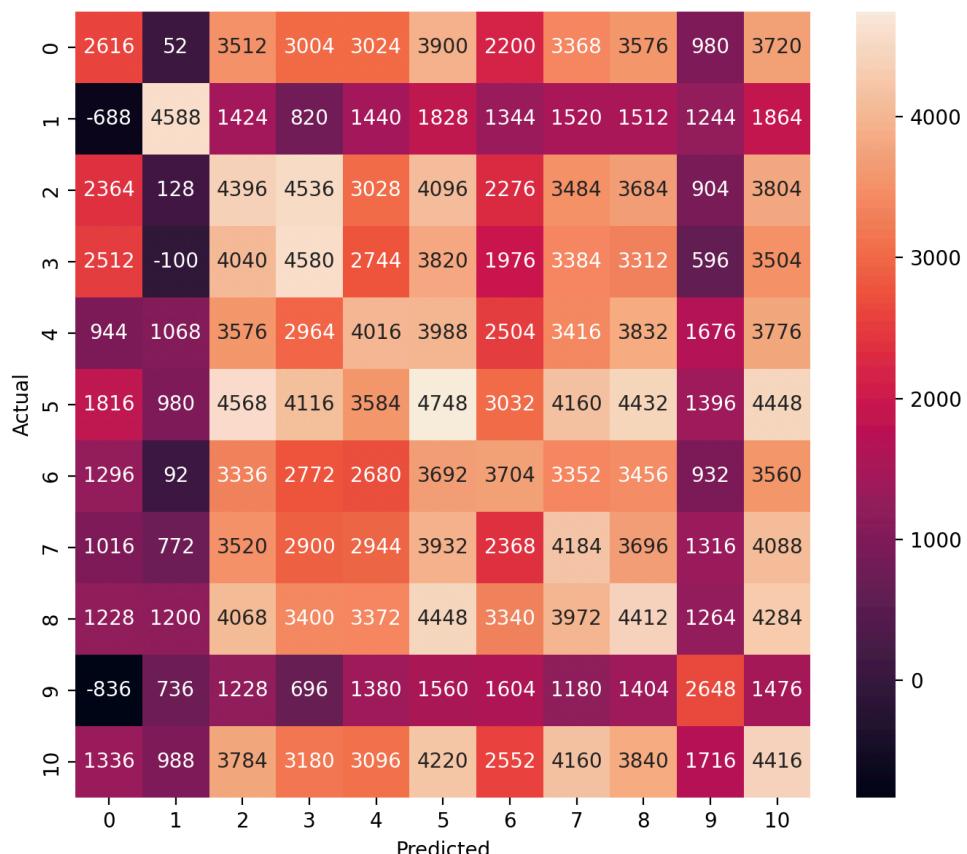
Interpretacija modela u kontekstu pronalaženja outlier-a i čestih pogrešaka može se provesti na različite načine, ovisno o vrsti modela i dostupnim podacima.

Jedan pristup može biti vizualizacija izlaza modela za primjere koji su klasificirani kao outlier-i ili koji često dovode do pogreške. Tako možemo vidjeti što model radi na takvim primjerima i pokušati razumjeti zašto dolazi do pogrešaka. Ova vizualizacija može uključivati različite tehnike, poput smanjenja dimenzionalnosti i projekcije podataka na 2D ili 3D prostor, ili grafova koji prikazuju aktivacije neurona ili težina.

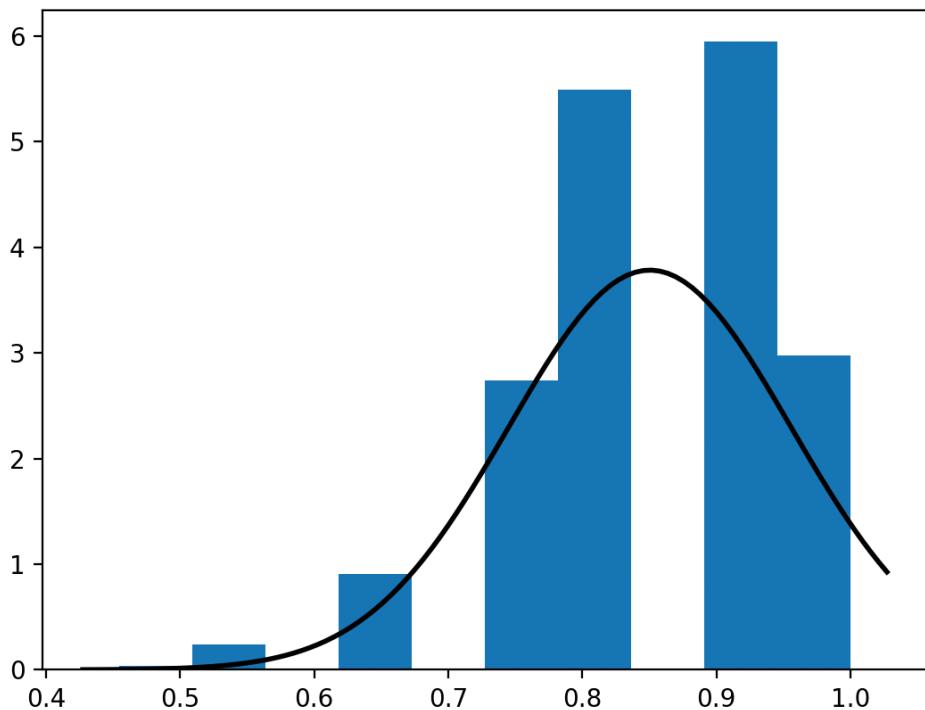
Drugi pristup može biti pronalaženje značajki koje su najviše doprinose pogreškama i outlier-ima. Ova analiza može uključivati metode poput LIME-a (Local Interpretable Model-Agnostic Explanations) ili SHAP-a (SHapley Additive exPlanations) koje omogućuju lokalno tumačenje modela. Tako možemo identificirati značajke koje su najvažnije za odluke modela u tim slučajevima i pokušati razumjeti zašto dovode do pogreške.

Također, možemo provesti analizu greške koja će nam dati uvid u distribuciju pogrešaka i kako se one raspoređuju u prostoru značajki. Tako možemo identificirati regije prostora značajki u kojima model loše radi i pokušati razumjeti zašto.

Konačno, možemo koristiti metode poput Grad-CAM (Gradient-weighted Class Activation Mapping) koji omogućuju vizualizaciju važnih područja u ulaznim slikama koji doprinose odlukama modela. Ova tehnika može nam pomoći da razumijemo koja su područja slike najvažnija za odluku modela i zašto.



Slika 6.5 Confusion matrix



**Slika 6.5** Distribucija Hamming scorea

Distribucija Hamming Scorea nad validacijskim podacima.  
Možemo vidjeti kako model uspješno detektira oko 85% puta.

Interesantno je preslušavati zvukove koji su na donjem rubu.

Tražeći outliere pri detekciji glazbenih instrumenata možemo doći do nekoliko zaključaka:

Moguće je da su neki instrumenti bolje prepoznati od drugih. Ako primijetimo da neki instrumenti često izazivaju outlier greške, to bi moglo značiti da naš model nije dovoljno dobro naučio prepoznavati te instrumente. U tom slučaju, možemo pokušati dodati više primjera tih instrumenata u naš skup podataka ili poboljšati način na koji model obrađuje te instrumente.

Outlier greške mogu se pojaviti ako postoje neobični ili neočekivani zvukovi u našem skupu podataka. Na primjer, ako se u skupu podataka pojavljuju zvukovi koji nisu povezani s glazbenim instrumentima, to bi moglo dovesti do outlier grešaka.

Outlier greške također mogu ukazivati na probleme u našem skupu podataka, kao što su pogreške u oznakama ili loše kvalitete audio zapisi. Ako primijetimo da određene greške često izlaze u istim audio datotekama, to bi moglo ukazivati na probleme s tim datotekama.

Također bismo trebali uzeti u obzir da su neke greške jednostavno prirodne i mogu se pojaviti čak i kod ljudi koji pokušavaju identificirati instrumente. To bi moglo biti zbog složenosti zvuka ili činjenice da neki instrumenti mogu zvučati slično.

## 8. Treniranje

U procesu stvaranja modela za prepoznavanje glazbenih instrumenata, važan korak je njegovo treniranje. Treniranje modela podrazumijeva proces učenja i podešavanja parametara u cilju postizanja željenih performansi modela. Treniranje modela zahtijeva veliku količinu računalnih resursa, a u ovom slučaju, korišteni su vast.ai i rtx 4090 grafička kartica.

Vast.ai je platforma za računalno oblak koja omogućava korisnicima pristup velikom broju virtualnih strojeva za treniranje i evaluaciju modela.

Korištenjem te platforme omogućeno nam je brzo i efikasno treniranje modela. Tijekom treniranja modela, koristili smo rtx 4090 grafičku karticu koja je trenutno jedna od najmoćnijih grafičkih kartica na tržištu. To nam je omogućilo da model treniramo brzo i učinkovito, a istovremeno smanjili vrijeme potrebno za postizanje dobrih performansi modela.

Uzimajući u obzir sve čimbenike i resurse koje smo koristili, model smo trenirali kroz nekoliko iteracija i optimizacija. Krajnji model postigao je izvrsne performanse u prepoznavanju glazbenih instrumenata, a u potpunosti je dostupan na našem GitHub rezervitoriju. Treniranje modela je kritičan korak u stvaranju učinkovitog sustava za prepoznavanje glazbenih instrumenata i zahtijeva veliku količinu truda i resursa.

Deployment je proces postavljanja aplikacije u produkciju, gdje korisnici mogu pristupiti i koristiti aplikaciju u stvarnom vremenu. U kontekstu ovih natuknica, deployment se odnosi na postavljanje modela strojnog učenja na web stranicu pomoću FastAPI i DigitalOcean platforme.

FastAPI je brzi, moderni web okvir otvorenog koda za izgradnju API-ja. Korištenjem FastAPI-a, lako je izgraditi API i povezati ga s različitim servisima. U ovom slučaju, FastAPI se koristi za postavljanje modela strojnog učenja na web stranicu.

DigitalOcean je platforma za upravljanje oblakom koja nudi VPS (Virtual Private Server) usluge. Koristeći DigitalOcean, moguće je jednostavno postaviti web aplikaciju u oblak.

Za ovaj projekt, korištena je Keras biblioteka za strojno učenje i treniranje modela. Trenirani model je potom ugrađen u FastAPI aplikaciju, koja je postavljena na DigitalOceanu. Nakon postavljanja aplikacije, korisnici mogu pristupiti modelu pomoću web stranice.

Postavljanje modela strojnog učenja pomoću FastAPI-a i DigitalOcean platforme omogućuje jednostavno i brzo postavljanje modela na web stranicu, što znači da korisnici mogu jednostavno pristupiti i koristiti model u stvarnom vremenu.

Github: "<https://github.com/ind1xa/LumenDataSci>"