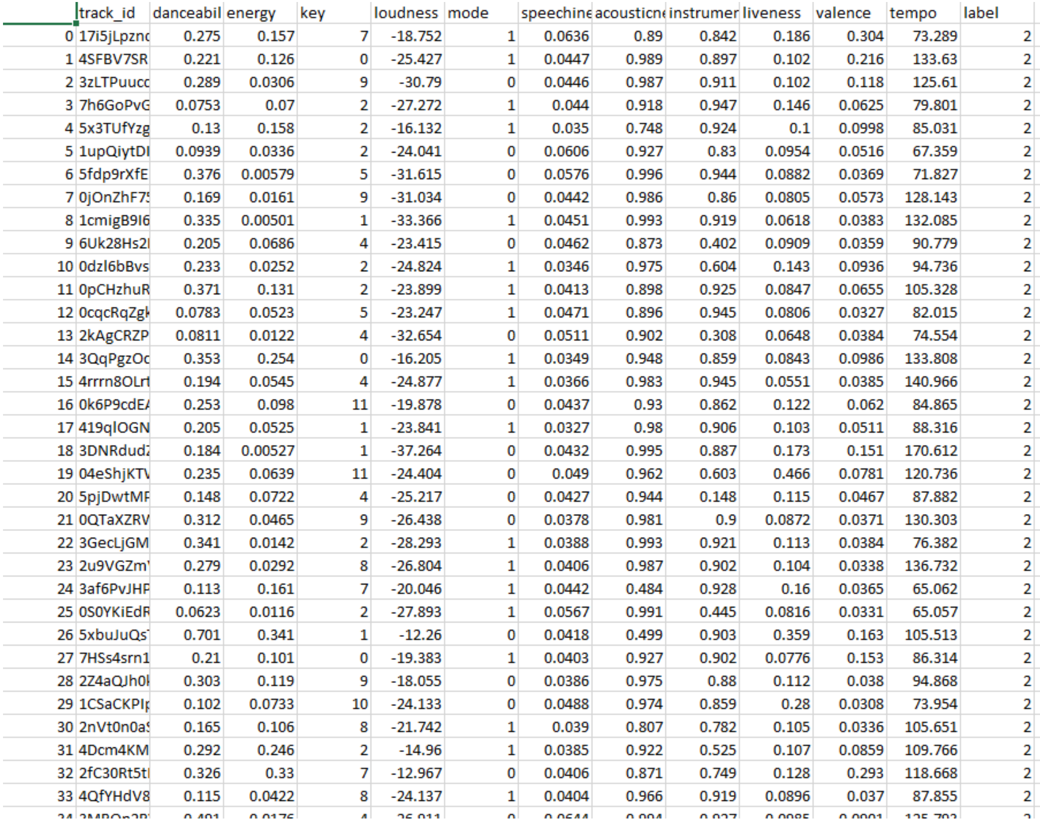
ECSE 484 - Assignment 10

Oleksii Fedorenko and Cameroon Byrne

April 16, 2022

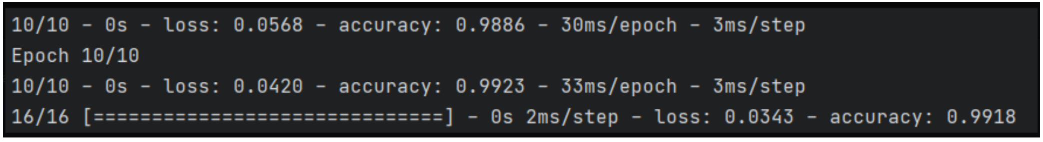
Have you ever listened to a song and wondered what genre it is? Or have your ever though about what genre of songs you like to listen to? We ary trying to create a neural network that would answer both of those questions. We are gathering data from the Spotify api about the song characteristics and then use that as training data to train a network to recognize the genre of songs that you like. The song features that we can gather from Spotify are: danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, and tempo. All of these values are floats and, combined together, are quite predictive of the song genre. We have already started designing a network that would be able to do the predictions for the genre.

Figure 1: Data format



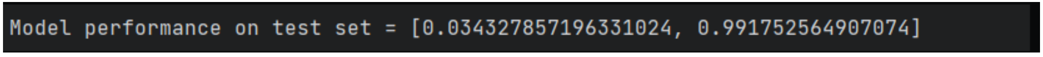
	track_id	danceabil	energy	key	loudness	mode	speechine	acousticn	instrumer	liveness	valence	tempo	label
0	17i5jLpznc	0.275	0.157	7	-18.752	1	0.0636	0.89	0.842	0.186	0.304	73.289	2
1	4SFBV7SR	0.221	0.126	0	-25.427	1	0.0447	0.989	0.897	0.102	0.216	133.63	2
2	3zLTPuucc	0.289	0.0306	9	-30.79	0	0.0446	0.987	0.911	0.102	0.118	125.61	2
3	7h6GoPvG	0.0753	0.07	2	-27.272	1	0.044	0.918	0.947	0.146	0.0625	79.801	2
4	5x3TUfYzg	0.13	0.158	2	-16.132	1	0.035	0.748	0.924	0.1	0.0998	85.031	2
5	1upQiytDI	0.0939	0.0336	2	-24.041	0	0.0606	0.927	0.83	0.0954	0.0516	67.359	2
6	5fdp9rXfE	0.376	0.00579	5	-31.615	0	0.0576	0.996	0.944	0.0882	0.0369	71.827	2
7	0jOnZhF7	0.169	0.0161	9	-31.034	0	0.0442	0.986	0.86	0.0805	0.0573	128.143	2
8	1cmigB9I6	0.335	0.00501	1	-33.366	1	0.0451	0.993	0.919	0.0618	0.0383	132.085	2
9	6Uk28Hs2I	0.205	0.0686	4	-23.415	0	0.0462	0.873	0.402	0.0909	0.0359	90.779	2
10	0dzl6bBvs	0.233	0.0252	2	-24.824	1	0.0346	0.975	0.604	0.143	0.0936	94.736	2
11	0pCHzhuR	0.371	0.131	2	-23.899	1	0.0413	0.898	0.925	0.0847	0.0655	105.328	2
12	0cqcRqZgk	0.0783	0.0523	5	-23.247	1	0.0471	0.896	0.945	0.0806	0.0327	82.015	2
13	2kAgCRZP	0.0811	0.0122	4	-32.654	0	0.0511	0.902	0.308	0.0648	0.0384	74.554	2
14	3QqPgzOc	0.353	0.254	0	-16.205	1	0.0349	0.948	0.859	0.0843	0.0986	133.808	2
15	4rrrn8OLrt	0.194	0.0545	4	-24.877	1	0.0366	0.983	0.945	0.0551	0.0385	140.966	2
16	0k6P9cdEA	0.253	0.098	11	-19.878	0	0.0437	0.93	0.862	0.122	0.062	84.865	2
17	419qlOGN	0.205	0.0525	1	-23.841	1	0.0327	0.98	0.906	0.103	0.0511	88.316	2
18	3DNRdudZ	0.184	0.00527	1	-37.264	0	0.0432	0.995	0.887	0.173	0.151	170.612	2
19	04eShjKTV	0.235	0.0639	11	-24.404	0	0.049	0.962	0.603	0.466	0.0781	120.736	2
20	5pjDwtMF	0.148	0.0722	4	-25.217	0	0.0427	0.944	0.148	0.115	0.0467	87.882	2
21	0QTaXZRV	0.312	0.0465	9	-26.438	0	0.0378	0.981	0.9	0.0872	0.0371	130.303	2
22	3GecLjGM	0.341	0.0142	2	-28.293	1	0.0388	0.993	0.921	0.113	0.0384	76.382	2
23	2u9VGZmY	0.279	0.0292	8	-26.804	1	0.0406	0.987	0.902	0.104	0.0338	136.732	2
24	3af6PvJHF	0.113	0.161	7	-20.046	1	0.0442	0.484	0.928	0.16	0.0365	65.062	2
25	0S0YKiEdR	0.0623	0.0116	2	-27.893	1	0.0567	0.991	0.445	0.0816	0.0331	65.057	2
26	5xbuJuQsT	0.701	0.341	1	-12.26	0	0.0418	0.499	0.903	0.359	0.163	105.513	2
27	7HSs4srn1	0.21	0.101	0	-19.383	1	0.0403	0.927	0.902	0.0776	0.153	86.314	2
28	2Z4aQJh0k	0.303	0.119	9	-18.055	0	0.0386	0.975	0.88	0.112	0.038	94.868	2
29	1CSaCKPIp	0.102	0.0733	10	-24.133	0	0.0488	0.974	0.859	0.28	0.0308	73.954	2
30	2nVt0n0aS	0.165	0.106	8	-21.742	1	0.039	0.807	0.782	0.105	0.0336	105.651	2
31	4Dcm4KM	0.292	0.246	2	-14.96	1	0.0385	0.922	0.525	0.107	0.0859	109.766	2
32	2fC30Rt5tI	0.326	0.33	7	-12.967	0	0.0406	0.871	0.749	0.128	0.293	118.668	2
33	4QfYHdV8	0.115	0.0422	8	-24.137	1	0.0404	0.966	0.919	0.0896	0.037	87.855	2
34	3MPOn2P	0.491	0.0176	4	-26.911	0	0.0644	0.994	0.927	0.0985	0.0901	125.793	2

Figure 2: Training set accuracy and loss



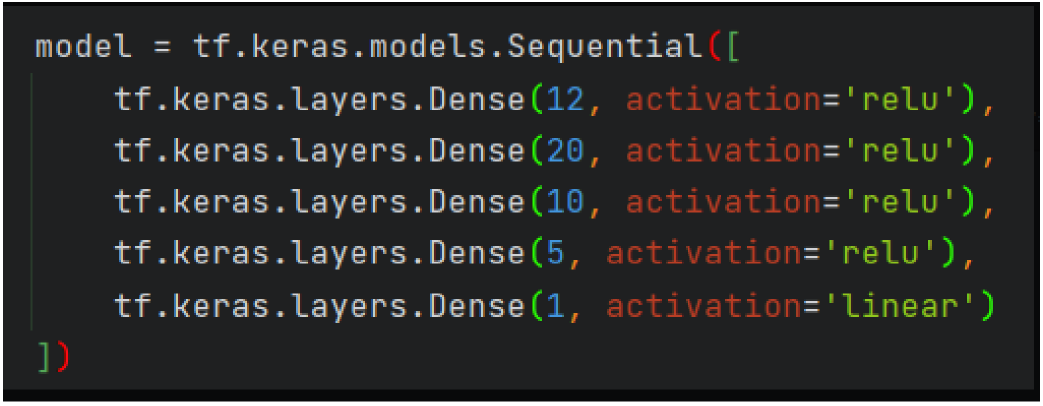
```
10/10 - 0s - loss: 0.0568 - accuracy: 0.9886 - 30ms/epoch - 3ms/step
Epoch 10/10
10/10 - 0s - loss: 0.0420 - accuracy: 0.9923 - 33ms/epoch - 3ms/step
16/16 [==============================] - 0s 2ms/step - loss: 0.0343 - accuracy: 0.9918
```

Figure 3: Testing set accuracy and loss



```
Model performance on test set = [0.034327857196331024, 0.991752564907074]
```

Figure 4: The model that we used



```
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(12, activation='relu'),
    tf.keras.layers.Dense(20, activation='relu'),
    tf.keras.layers.Dense(10, activation='relu'),
    tf.keras.layers.Dense(5, activation='relu'),
    tf.keras.layers.Dense(1, activation='linear')
])
```

But we would not stop there. There are a couple of ideas we have in mind about how we can apply this data. For examply we can get a sample of songs from your favorite artist and album and determine their genre and mood. We would also try to make a neural network that would train on the data of the songs you like and a score you assign to train the network, rating the songs by how much you might like them and producing a ranking based on the predicted score for all of them.