پاسخ سوال سوم تمرین دوم هوش مصنوعی

گزارش مراحل تحلیل و پیش‌پردازش داده‌ها و همچنین ارائه مدلی برای تخمین انرژی

الف) اضافه کردن کتابخانه‌های مورد نیاز و خواندن داده‌ها به کمک Pandas

نخست تمامی کتابخانه‌های لازم را وارد می‌کنیم. پس از آن باید با استفاده از تابع read_csv() داده‌ها را از فایل csv خواند. در نتیجه این کار فایل به فرمت DataFrame درخواهد آمد. با استفاده از تابع head() می‌توان چند سطر اول این مجموعه داده را مشاهده کرد.

	track_id	disc_number	duration_ms	explicit	track_name	track_name_farsi	artist_name	artist_name_farsi	popularity	track_number	...	spe
0	31iPeC6l0AiRW8lnOxNKzm	1	446880	False	Ghazale Taze	NaN	Salar Aghili	سالار عقیلی	NaN	1	...	
1	4Fi46ha8teWYTwk0b8fNPi	1	851920	False	Ayeeneye Hosn	NaN	Salar Aghili	سالار عقیلی	NaN	2	...	
2	0lQAe6EslKA7CUsS7SCW6Q	1	293160	False	Tarke Eshgh	NaN	Salar Aghili	سالار عقیلی	NaN	3	...	
3	6dAFmJdVsKk5ksCpGqnKgO	1	648720	False	Moghbacheye Bade Foroosh	NaN	Salar Aghili	سالار عقیلی	NaN	4	...	
4	4VSDJGyEdSMB8UL4fDSCvv	1	273480	False	Bigharar	NaN	Salar Aghili	سالار عقیلی	NaN	5	...	

5 rows × 32 columns

ب) شناسایی داده‌های از دست رفته

به سادگی می‌توان با استفاده از متد isna() داده‌های از دست رفته را شناسایی کرد و به کمک sum() تعداد کل missing valueها را برای هر ستون از داده‌ها محاسبه کرد.

همانطور که از نتایج مشخص است ویژگی‌های track_name_farsi، popularity، album_href، album_total_tracks، key_name، mode_name، key_mode دارای داده از دست رفته هستند. برای ۳ ویژگی key_name، mode_name، key_mode از آنجایی که تعداد ناچیزی از آن‌ها ناموجود هستند، به سادگی می‌توان از آن‌ها صرف نظر کرد بدون آنکه بر روی نتایج اثر منفی بگذارد. پس برای هر ویژگی تاپل‌های null را از طریق متد isnull() شناسایی کرده و به کمک drop() آن‌ها را از دیتاست حذف می‌کنیم. اما برای ویژگی‌های track_name_farsi، popularity، album_href، album_total_tracks از آن جایی که بیش از ۹۰ درصد داده‌های در هر ستون از دست رفته است، مجبور به حذف ویژگی هستیم.

ج) تمیزسازی داده‌ها

این کار را با حذف ستون album_id، track_id و track_href از داده‌ها آغاز می‌کنیم. چرا که این ویژگی‌ها نامرتبط به پردازش‌های آینده تلقی می شوند و برای آموزش مدل به آن‌ها نیازی نداریم. در ادامه لازم است مقادیر ویژگی duration_ms از میلی‌ثانیه به ثانیه تبدیل شوند تا ناسازگاری بین داده‌ها ایجاد نشود.

سپس باید داده‌های دسته‌بندی شده یا همان categorical را به مقادیر عددی تبدیل کنیم که برای اینکار از label encoder استفاده شده است. شمای نهایی داده‌ها به شکل زیر خواهد بود.

	disc_number	duration_ms	explicit	track_name	artist_name	artist_name_farsi	track_number	album_name	album_release_date	album_release_year	...	speech
0	1	446.88	0	2706	58	24	1	1945	1543	2020	...	(
1	1	851.92	0	593	58	24	2	1945	1543	2020	...	(
2	1	293.16	0	6484	58	24	3	1945	1543	2020	...	(
3	1	648.72	0	4469	58	24	4	1945	1543	2020	...	(
4	1	273.48	0	1191	58	24	5	1945	1543	2020	...	(

5 rows × 25 columns

در نهایت ویژگی جدیدی به داده‌ها اضافه می‌کنیم که نشان‌دهنده دهه انتشار یک آلبوم است تا در ادامه بتوانیم قطعات را برحسب دهه انتشارشان بررسی کنیم. برای این‌کار از تابع year_to_decade استفاده شده است.

د) انجام EDA و Visualization برای بدست آوردن بینش از داده‌ها

ابتدا برای بدست آوردن اطلاعات آماری این مجموعه داده، می‌توان با استفاده از تابع describe() توصیفی آماری از متغیرهای عددی مشاهده کرد.

	disc_number	duration_ms	explicit	track_name	artist_name	artist_name_farsi	track_number	album_name	album_release_date	album_release_ye
count	10488.000000	10488.000000	10488.000000	10488.000000	10488.000000	10488.000000	10488.000000	10488.000000	10488.000000	10488.0000
mean	1.070080	287.975313	0.002002	3701.980740	34.884058	33.586575	5.593154	1043.473875	578.168764	2008.8959
std	0.388827	169.926950	0.044704	2156.224016	20.140734	18.018821	4.544148	617.791304	458.752226	8.7260
min	1.000000	3.996000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1974.0000
25%	1.000000	203.440000	0.000000	1867.000000	15.000000	17.000000	2.000000	492.000000	191.000000	2005.0000
50%	1.000000	252.988500	0.000000	3690.000000	38.000000	33.000000	5.000000	1092.000000	387.000000	2011.0000
75%	1.000000	331.840000	0.000000	5533.250000	53.000000	50.000000	8.000000	1600.000000	973.000000	2016.0000
max	4.000000	3978.450000	1.000000	7552.000000	68.000000	68.000000	32.000000	2072.000000	1571.000000	2020.0000

8 rows × 26 columns

همچنین با استفاده از تابع info() و shape می‌توان تعداد سطرها و ستون‌های داده به همراه نوع آن‌ها را مشاهده کرد که نتایج آن در نوت بوک موجود است.

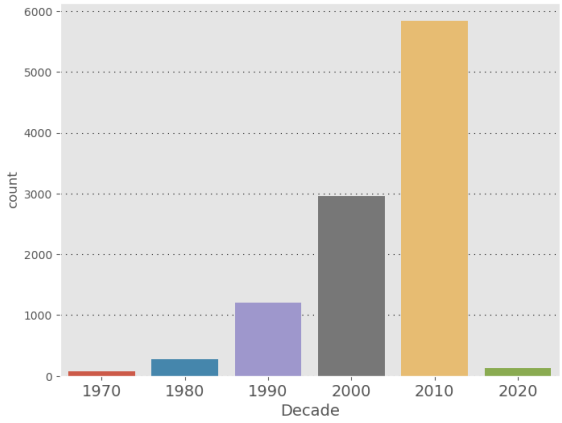
```
df.shape
```

```
(10488, 26)
```

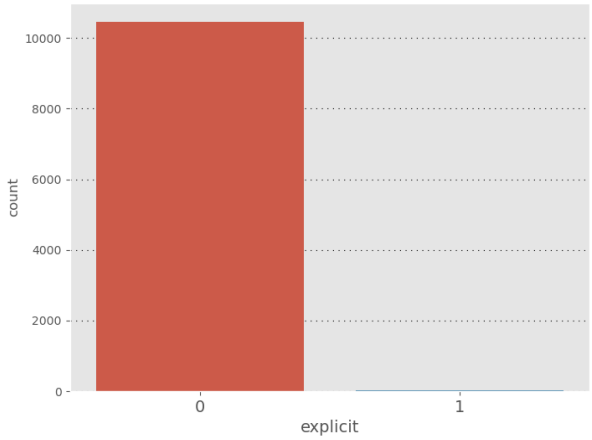
سپس به سراغ رسم نمودارها و data visualization می‌رویم.

نمودار توزیع

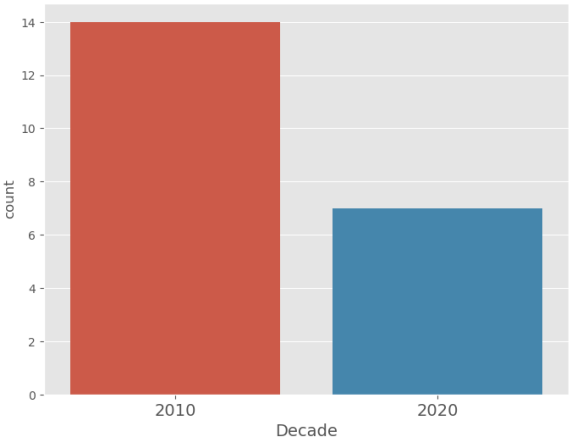
در ادامه با رسم نمودارهایی تعداد مشاهدات در هر دسته از ویژگی‌ها را تصویرسازی می‌کنیم تا دید بهتری نسبت به داده‌ها داشته باشیم.



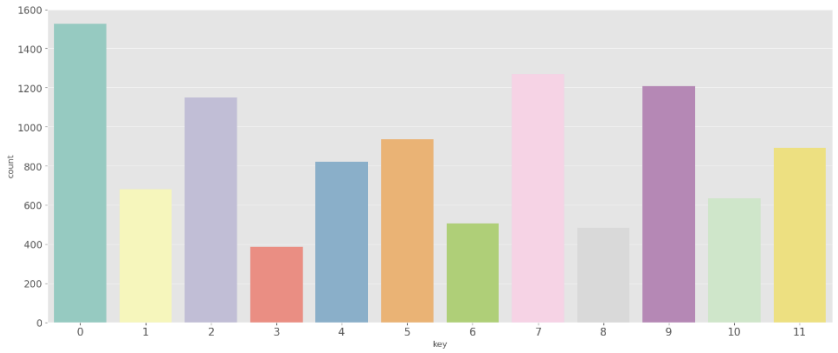
نمودار ۱: بیش‌تر قطعات موسیقی این مجموعه داده در دهه‌های ۲۰۰۰ و ۲۰۱۰ منتشر شده‌اند.



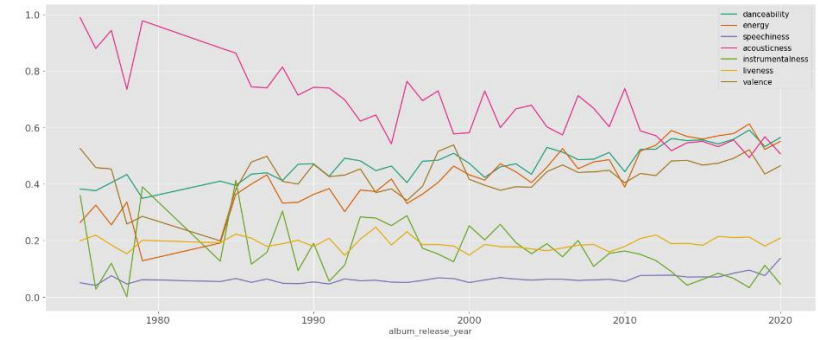
نمودار ۲: اکثر قطعات ماهیت خشونت‌آمیز ندارند.



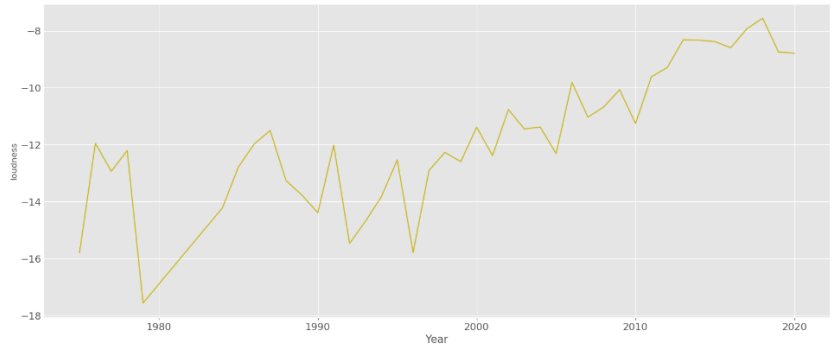
نمودار ۳: قطعات خشونت‌آمیز در دهه‌های ۲۰۱۰ و ۲۰۲۰ منتشر شده‌اند.



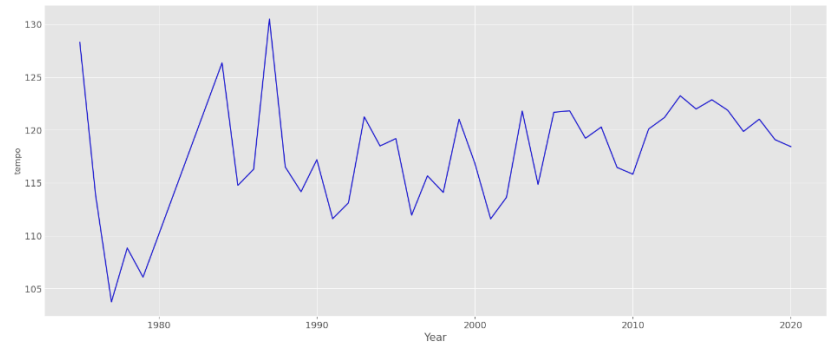
نمودار ۴: فراوانی قطعات بر حسب کلید



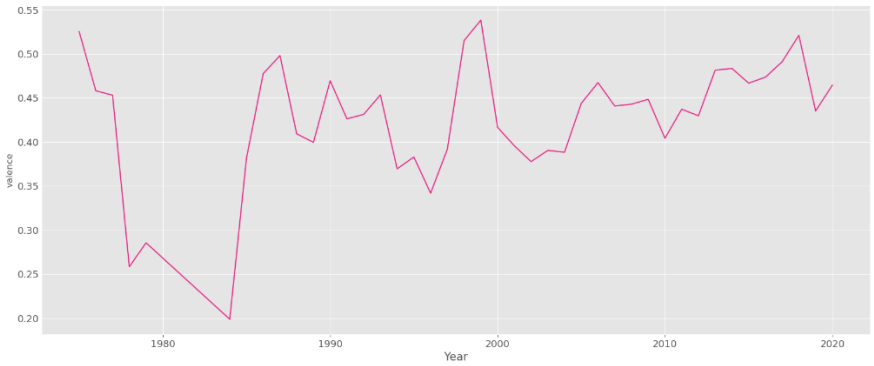
نمودار ۵: روند تغییرات ویژگی‌های موسیقیایی قطعات در طول زمان



نمودار ۶: روند تغییرات میزان بلندی صدای قطعات در طول زمان



نمودار ۷: روند تغییرات میزان ضرب آهنگ قطعات در طول زمان

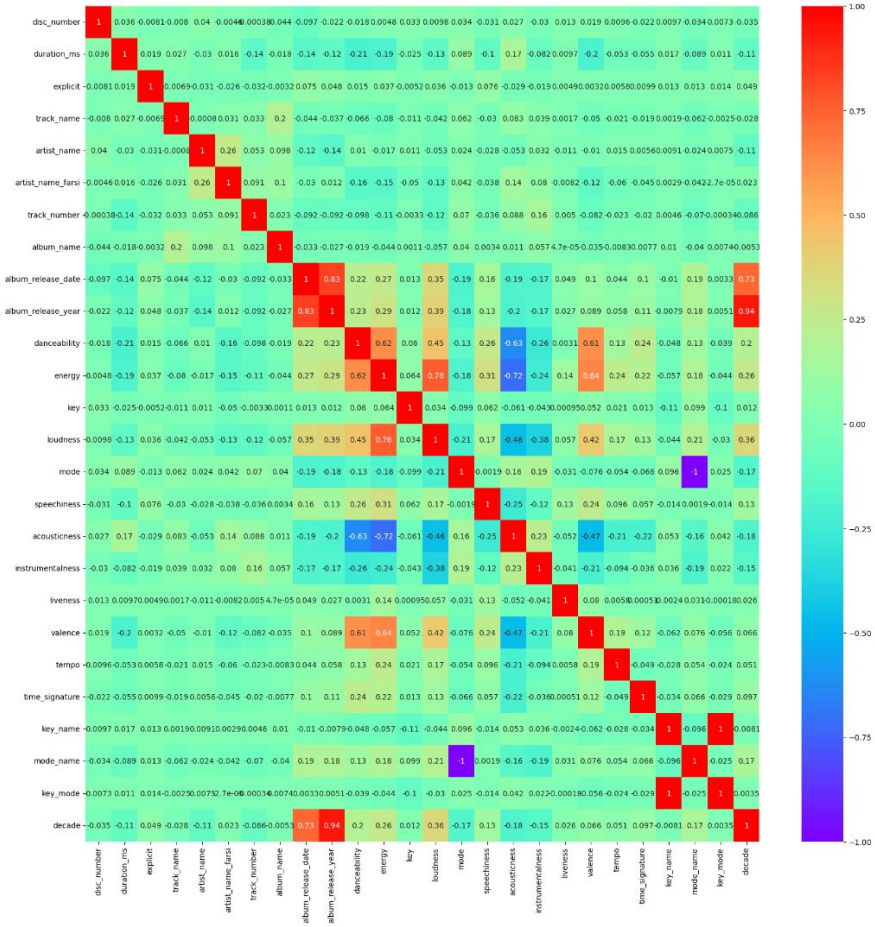


نمودار ۸: روند تغییرات شاد یا غمگین بودن قطعات در طول زمان

به همین ترتیب تمامی نمودارها در نوت‌بوک ترسیم شده‌اند.

ه) رسم ماتریس هبستگی

پیش‌تر داده‌های دسته‌بندی شده یا همان categorical را به داده‌های عددی تبدیل کردیم. حال با رسم نمودار هبستگی میزان همبستگی دوبه‌دوی متغیرها به یکدیگر را نمایش می‌دهیم.



نمودار ۹

و) تقسیم داده‌ها به مجموعه دادهای آموزشی و تستی

در این مرحله می‌خواهیم با استفاده از داده‌ها مدلی برای تخمین انرژی قطعات ارائه کنیم. در نتیجه ویژگی energy به عنوان برچسب y و بقیه داده‌ها به عنوان داده‌های ورودی محسوب می‌شوند. ابتدا داده‌ها را به کمک ابزار shuffle مخلوط می‌کنیم. سپس داده‌ها را به X train، X test، y train و y test با نسبت ۰.۲ تقسیم می‌کنیم.

```
# Check the shape of X_train and X_test

X_train.shape, X_test.shape
```

```
((8390, 25), (2098, 25))
```

در همین مرحله مقیاس‌بندی ویژگی‌ها را نیز انجام می‌دهیم؛ چراکه آموزش یک شبکه MLP و همچنین مدل رگرسیون ، به آن حساس است. برای اینکار از ماژول StandardScaler() استفاده می‌کنیم.

ی) آموزش مدل

در نهایت می‌خواهیم با دو روش رگرسیون خطی و روش‌های بر پایه mlp مدل را آموزش دهیم. ابتدا به آموزش مدل با استفاده از رگرسیون خطی می‌پردازیم. با استفاده از ابزار LinearRegression از sklearn.linear_model به سادگی می‌توان با ساخت یک شی از آن و بکارگیری متد fit() مدل را آموزش داد. در نهایت نیز به کمک متد score() دقت مدل تعلیم داده شده محاسبه می‌شود.

```
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred_test = lr.predict(X_test)
LinearRegressionScore = lr.score(X_test,y_test)
LinearRegressionScore
```

```
0.8274039051693888
```

برای تشخیص اینکه بیش برازش و یا کم برازش رخ داده است یا خیر، دقت را روی داده‌های تست و داده‌های آموزشی محاسبه می‌کنیم. با توجه به آنکه دقت هر دو متناسب با یکدیگر بدست آمده است، نتیجه می‌گیریم مدل به خوبی آموزش دیده است.

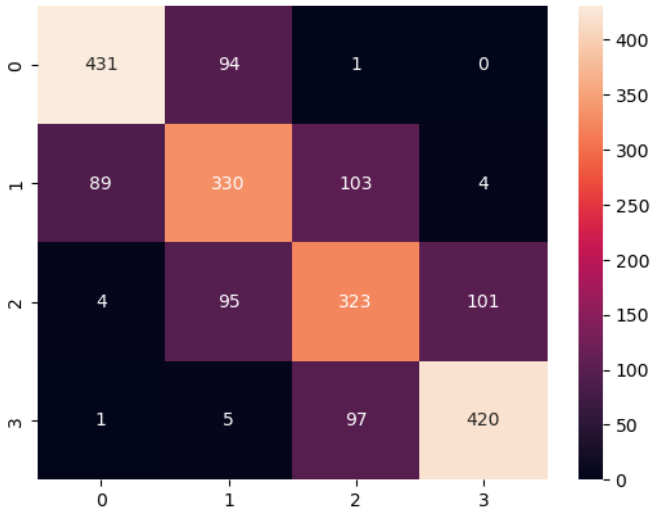
```
# Check for overfitting and underfitting

print("score on train data: ", lr.score(X_train, y_train))
print("score on test data: ", lr.score(X_test, y_test))
```

```
score on train data:  0.8285823297337059
score on test data:  0.8274039051693888
```

در ادامه ماتریس درهم ریختگی ترسیم و معیارهای recall، precision و accuracy محاسبه شده‌اند. از آنجایی که ماتریس درهم ریختگی ابزاری برای اندازه‌گیری کارایی مسائل دسته‌بندی به کمک یادگیری ماشینی است، ابتدا لازم است نتایج پیش‌بینی و تست بازه بندی شوند که در اینجا با استفاده از متد qcut() در ۴ دسته، طبقه‌بندی شده‌اند. رسم نمودار بر روی مقادیر جدید، مانند ترسیم ماتریس درهم ریختگی بر روی چندین کلاس در مسائل دسته‌بندی خواهد بود.

	precision	recall	f1-score	support
0	0.82	0.82	0.82	526
1	0.63	0.63	0.63	526
2	0.62	0.62	0.62	523
3	0.80	0.80	0.80	523
accuracy			0.72	2098
macro avg	0.72	0.72	0.72	2098
weighted avg	0.72	0.72	0.72	2098



نمودار ۱۰

ماتریس درهم ریختگی بالا، مقایسه‌ای از مقادیر پیش‌بینی شده و مقادیر واقعی را نشان می‌دهد. مدلی که به خوبی آموزش دیده باشد باید در هر دسته بیش‌ترین نرخ پیش‌بینی درست را فراهم کند که با توجه به نمودار بالا، این شرط برآورده شده است و اکثر پیش‌بینی‌ها با مقدار واقعی داده‌ها تطابق دارند (قطر ماتریس نشان‌گر این مطلب است.)

از دیگر معیارهای ارزشیابی مدل‌های دسته‌بندی، مقادیر recall، precision و accuracy هستند. در اینجا چون مدل اکثر پیش‌بینی‌ها را به درستی انجام داده است، مقدار هر سه معیار ۰.۷۱ است که نتیجه نسبتاً مطلوبی است.

```
precision_recall_fscore_support(y_test_disc, y_pred_test_disc, average='micro')
```

```
(0.7168732125834127, 0.7168732125834127, 0.7168732125834127, None)
```

همچنین برای این مسئله که یک مسئله رگرسیون است معیارهای Mean Absolute Error و Mean Squared Error تعریف می‌شوند که مقدار میانگین خطای مطلق یک مدل با توجه به مجموعه آزمایشی، میانگین مقادیر مطلق خطاهای پیش بینی منفرد در تمام نمونه‌های مجموعه آزمایشی خواهد بود. مقدار ۰.۰۷ برای مجموعه داده مورد بررسی مناسب به نظر می‌رسد. میانگین مربعات خطا، اندازه‌گیری میزان نزدیکی خط رگرسیون به مجموعه‌ای از نقاط داده است. هر چه این مقدار کمتر باشد، مدل بهتر تلقی می شود.

```
Mean Absolute Error: 0.079233430960367
Mean Squared Error: 0.010514384734059065151789980064
Root Mean Squared Error: 0.102539673951398
```

برای آموزش مدل بر پایه MLP از MLPRegressor در sklearn.neural_network استفاده شده است.این ماژول با دریافت تعداد لایه‌های پنهان، تابع فعال‌سازی و random state یک شی ساخته و فراخوانی متدهای fit و predict و در نهایت score تمام عملیات آموزش و محاسبه دقت مدل را انجام می‌دهد. پارامترهای در نظر گرفته شده در این تمرین به شرح زیر است:

hidden_layer_sizes=(60, 50), activation = 'tanh', solver='adam', random_state=98

حدس اولیه برای پارامترهای این مساله hidden_layer_sizes = (150, 100) با استفاده از تابع فعال‌سازی relu بود. اما به دلیل رخ دادن بیش برازش نتیجه‌ی مطلوبی حاصل نشد. پس از آزمون و خطا و با توجه به بازخوردی که از مدل دریافت می‌شد، در نهایت مشخص شد که مدل با استفاده از تابع فعال سازی tanh به طرز چشم‌گیری بهبود دقت دارد. همینطور با چندین مرحله تغییر hidden_layes_sizes مقدار فعلی به دست آمد که دقتی برابر ۰.۸۴ برای مدل به همراه داشت. همچنین در ابتدا ۳۰۰ تکرار برای آموزش مدل در نظر گرفته شده بود، اما با رسم نمودار loss curve مشخص شد مدل با مقدار پیش‌فرض به خوبی تعلیم می‌بیند.

```
mlp_reg = MLPRegressor(hidden_layer_sizes=(60, 50), activation = 'tanh', solver='adam', random_state=98)
mlp_reg.fit(X_train, y_train)
ypred = mlp_reg.predict(X_test)
MLPRegressorScore = mlp_reg.score(X_test, y_test)
MLPRegressorScore
```

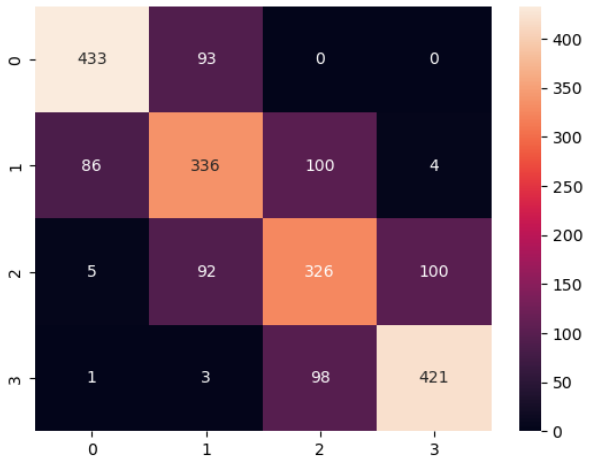
```
0.8487496362883175
```

همچنین در ادامه محاسباتی برای تشخیص بیش برازش و کم برازش صورت گرفته است که نشان می‌دهد این دو مشکل رایج در مدل تعلیم دیده با مقادیر فعلی وجود ندارند. برای تشخیص اینکه بیش برازش و یا کم برازش رخ داده است یا خیر، دقت را روی داده‌های تست و داده‌های آموزشی محاسبه می‌کنیم. با توجه به آنکه دقت هر دو متناسب با یکدیگر بدست آمده است، نتیجه می‌گیریم مدل به خوبی آموزش دیده است.

```
score on train data:  0.8702649564454867
score on test data:  0.8487496362883175
```

در ادامه ماتریس درهم ریختگی ترسیم و معیارهای recall، precision و accuracy محاسبه شده‌اند. از آنجایی که ماتریس درهم ریختگی ابزاری برای اندازه‌گیری کارایی مسائل دسته‌بندی به کمک یادگیری ماشینی است، ابتدا لازم است نتایج پیش‌بینی و تست بازه بندی شوند که در اینجا با استفاده از متد qcut() در ۴ دسته، طبقه‌بندی شده‌اند. رسم نمودار بر روی مقادیر جدید، مانند ترسیم ماتریس درهم ریختگی بر روی چندین کلاس در مسائل دسته‌بندی خواهد بود.

	precision	recall	f1-score	support
0	0.82	0.82	0.82	526
1	0.64	0.64	0.64	526
2	0.62	0.62	0.62	523
3	0.80	0.80	0.80	523
accuracy			0.72	2098
macro avg	0.72	0.72	0.72	2098
weighted avg	0.72	0.72	0.72	2098



نمودار ۱۱

ماتریس درهم ریختگی بالا، مقایسه‌ای از مقادیر پیش‌بینی شده و مقادیر واقعی را نشان می‌دهد. مدلی که به خوبی آموزش دیده باشد باید در هر دسته بیش‌ترین نرخ پیش‌بینی درست را فراهم کند که با توجه به نمودار بالا، این شرط برآورده شده است و با تقریب خوبی اکثر پیش‌بینی‌ها با مقدار واقعی داده‌ها تطابق دارند (قطر ماتریس نشان‌گر این مطلب است.)

از دیگر معیارهای ارزشیابی مدل‌های دسته‌بندی، مقادیر recall، precision و accuracy هستند. در اینجا چون مدل اکثر پیش‌بینی‌ها را به درستی انجام داده است، مقدار هر سه معیار ۰.۷۲ است که نتیجه تقریبا مطلوبی است.

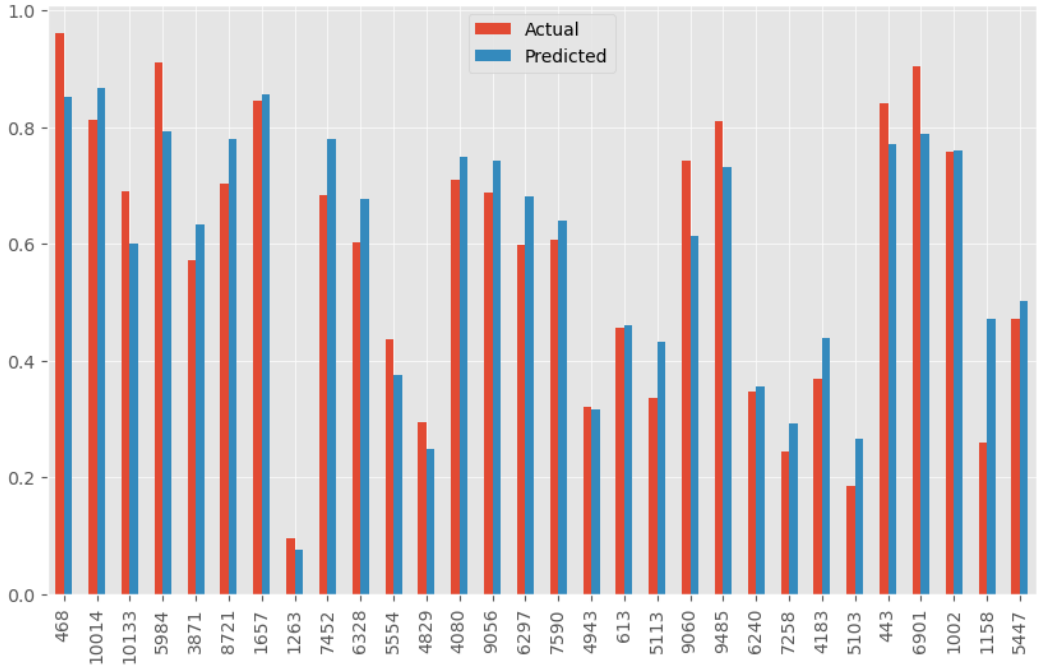
```
precision_recall_fscore_support(y_test_disc, y_pred_test_disc, average='micro')
```

```
(0.7225929456625357, 0.7225929456625357, 0.7225929456625356, None)
```

همچنین برای این مسئله که یک مسئله رگرسیون است معیارهای Mean Absolute Error و Mean Squared Error تعریف می‌شوند که مقدار میانگین خطای مطلق یک مدل با توجه به مجموعه آزمایشی، میانگین مقادیر مطلق خطاهای پیش بینی منفرد در تمام نمونه‌های مجموعه آزمایشی خواهد بود. مقدار ۰.۰۷ برای مجموعه داده مورد بررسی مناسب به نظر می‌رسد. میانگین مربعات خطا، اندازه‌گیری میزان نزدیکی خط رگرسیون به مجموعه‌ای از نقاط داده است. هر چه این مقدار کمتر باشد، مدل بهتر تلقی می شود.

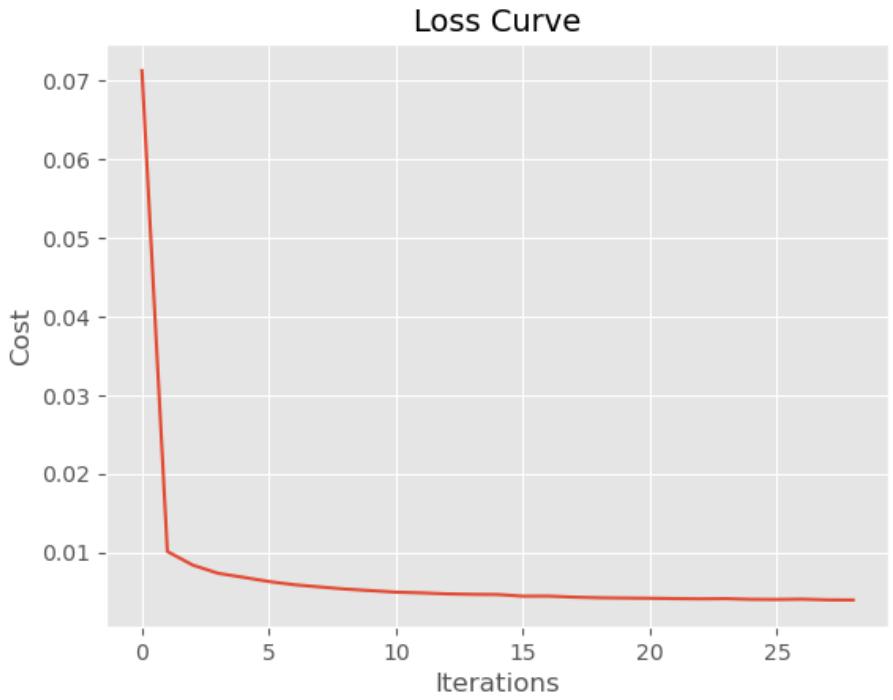
```
Mean Absolute Error: 0.0726400411212166
Mean Squared Error: 0.009015096109835164
Root Mean Squared Error: 0.09494785995395137
```

در ادامه نموداری رسم شده است که میزان دقت تخمین‌های صورت گرفته توسط مدل را بر روی نمونه‌ای ۳۰ عددی نشان می‌دهد که همگی بسیار نزدیک به مقدار اصلی هستند.



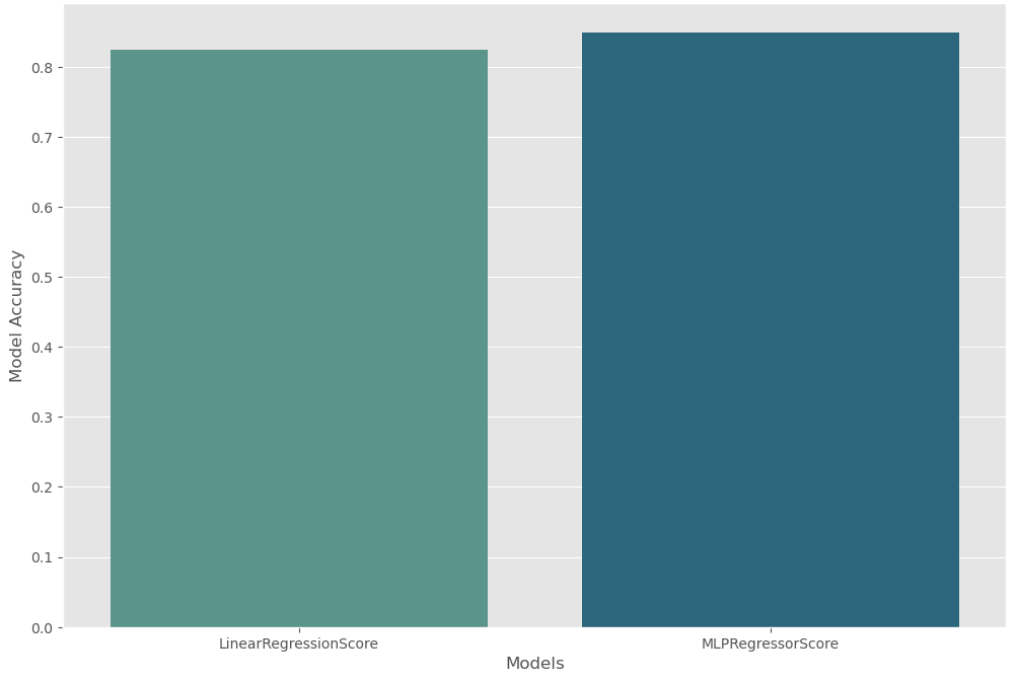
نمودار ۱۲

در ادامه نمودار loss curve ترسیم شده است که روند کاهشی میزان خطا در تکرارهای متوالی را نشان می‌دهد.



نمودار ۱۳

در نهایت می‌توان نتیجه آموزش با رگرسیون خطی و رگرسیون بر پایه MLP را در نمودار زیر مشاهده کرد.



نمودار ۱۴