پاسخ سوال اول تمرین دوم هوش مصنوعی

گزارش مراحل تحلیل و پیش‌پردازش داده‌ها و همچنین ارائه مدلی برای تخمین نمره پروژه

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.

	Unnamed: 0	ID	uni	DS	ALG	ML	DL	DB	HW	OS	NTW	DSP	STandLAL	prj
0	0	974285.0	Shiraz UniVersity	4.613070	19.219966	13.777694	1.768949	12.918400	12.560763	17.544232	6.242864	1.286421	18.607406	57.462136
1	1	563921.0	NaN	NaN	10.416559	10.294882	10.370097	0.236130	9.978527	16.255803	11.639155	17.540606	0.664260	52.820606
2	2	308307.0	Isfahan University of Tecchnology	14.193912	19.516232	8.193591	9.325052	19.036204	9.339919	13.861366	14.092048	1.233343	9.441342	67.046251
3	3	NaN	Tabriz University	16.422105	9.990100	5.491504	7.267110	14.041443	8.326088	15.791592	8.534165	9.126567	6.920051	55.419282
4	4	570319.0	Shiraz UniVersity	13.732369	4.311718	8.025760	5.766623	8.488922	2.124606	4.632900	10.456614	7.636695	11.098141	43.273511

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.

	missing value count
Unnamed: 0	0
ID	19
uni	36
DS	16
ALG	10
ML	1
DL	1
DB	1
HW	3
OS	24
NTW	1
DSP	1
STandLAL	2
prj	2

همانطور که از نتایج مشخص است در همه‌ی ستون‌ها تعداد ناچیزی داده از دست رفته موجود است. به دلیل آنکه تعداد کل داده های از دست رفته برای هر کدام از ویژگی‌های این مجموعه داده کم است، 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.

ج) تمیزسازی داده‌ها

این کار را با حذف ستون ID و Unnamed: 0 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 تمام مقادیر را برای ستون uni از نظر حروف بزرگ و کوچک یکدست کنیم که برای اینکار از str.lower() استفاده شده است.

همانطور که در صورت سوال گفته شده، برخی دانشگاه‌هایی که در این مجموعه داده وجود دارد یک بار با نام کامل و یک بار با سرنام معرفی شده‌اند. پس این مقادیر را به کمک متد replace() جایگزین کرده تا تمامی دانشگاه‌ها با نام کامل خود در مجموعه داده‌ها موجود باشند. در نهایت 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.

	uni	DS	ALG	ML	DL	DB	HW	OS	NTW	DSP	STandLAL	prj
0	6	4.613070	19.219966	13.777694	1.768949	12.918400	12.560763	17.544232	6.242864	1.286421	18.607406	57.462136
2	3	14.193912	19.516232	8.193591	9.325052	19.036204	9.339919	13.861366	14.092048	1.233343	9.441342	67.046251
3	7	16.422105	9.990100	5.491504	7.267110	14.041443	8.326088	15.791592	8.534165	9.126567	6.920051	55.419282
4	6	13.732369	4.311718	8.025760	5.766623	8.488922	2.124606	4.632900	10.456614	7.636695	11.098141	43.273511
6	8	12.627720	17.081490	2.486638	10.361988	11.640996	11.085180	17.306854	7.883087	17.600587	18.884534	64.087750

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.

	uni	DS	ALG	ML	DL	DB	HW	OS	NTW	DSP	STandLAL	prj
count	915.000000	915.000000	915.000000	915.000000	915.000000	915.000000	915.000000	915.000000	915.000000	915.000000	915.000000	915.000000
mean	5.069945	9.995207	10.044200	9.800999	9.766902	10.319046	10.204428	10.101166	10.069689	9.951115	10.360928	56.388549
std	2.326595	5.731800	5.826290	5.703610	5.855642	5.857843	7.566368	6.873874	5.631195	5.756028	5.791537	11.522999
min	0.000000	0.011261	0.055391	0.028951	0.020001	0.000727	-7.390506	-17.611152	0.013312	0.012463	0.013459	17.717730
25%	3.000000	5.081457	5.040771	4.984407	4.835035	5.326907	5.130636	5.058178	5.406178	5.110303	5.264572	48.389906
50%	6.000000	10.022238	9.926055	9.851798	9.725941	10.087395	9.794095	10.335979	10.097890	9.851861	10.222342	56.201281
75%	7.000000	14.858347	15.099152	14.499090	14.413309	15.564909	15.123588	15.090184	15.083018	15.047385	15.422406	63.845680
max	10.000000	19.974687	19.931012	19.971160	55.236815	19.997719	114.879538	113.497786	19.934011	19.927692	19.941939	91.853542

همچنین با استفاده از تابع info() و shape می‌توان تعداد سطرها و ستون‌های داده به همراه نوع آن‌ها را مشاهده کرد.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 915 entries, 0 to 999
Data columns (total 12 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   uni       915 non-null    int32
 1   DS        915 non-null    float64
 2   ALG       915 non-null    float64
 3   ML        915 non-null    float64
 4   DL        915 non-null    float64
 5   DB        915 non-null    float64
 6   HW        915 non-null    float64
 7   OS        915 non-null    float64
 8   NTW       915 non-null    float64
 9   DSP       915 non-null    float64
 10  STandLAL  915 non-null    float64
 11  prj       915 non-null    float64
dtypes: float64(11), int32(1)
memory usage: 89.4 KB
```

```
df.shape
```

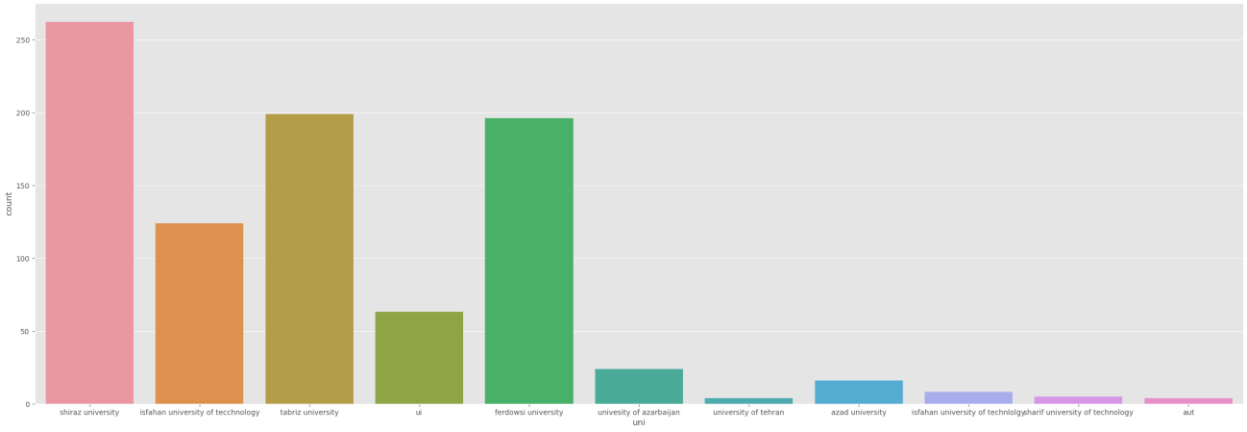
```
(915, 12)
```

همانطور که از نتیجه تحلیل آماری مشخص است در نمرات دروس سیستم عامل، یادگیری عمیق و سخت‌افزار outlier وجود دارد. از آنجایی که نمرات دروس باید عددی بین ۰ تا ۲۰ باشند، تاپل‌هایی که نمره‌ای خارج از این بازه برای آن‌ها ثبت شده است (که در مجموع ۱۰ عدد هستند) را شناسایی کرده و از مجموعه داده حذف می‌کنیم.

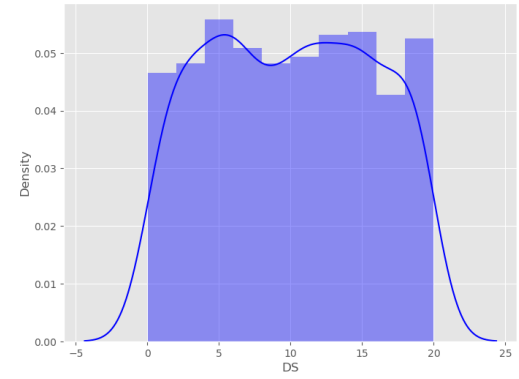
سپس به سراغ رسم نمودارها و data visualization می‌رویم.

نمودار توزیع

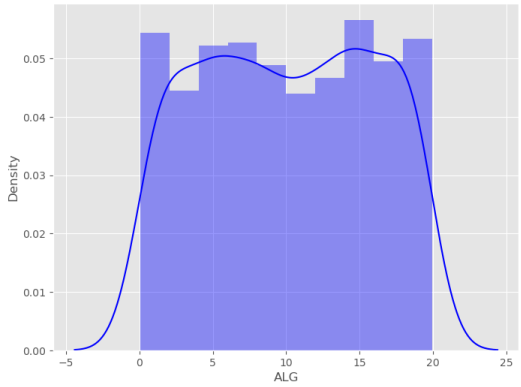
در ادامه با رسم نمودارهایی تعداد مشاهدات در هر دسته از ویژگی‌ها را تصویرسازی می‌کنیم تا دید بهتری نسبت به داده‌ها داشته باشیم.



نمودار ۱: داده‌های موجود اکثراً از دانشگاه‌های شیراز، تبریز و فردوسی مشهد هستند.



نمودار۲: توزیع نمرات درس ساختمان داده با میانگین ۹.۹۹

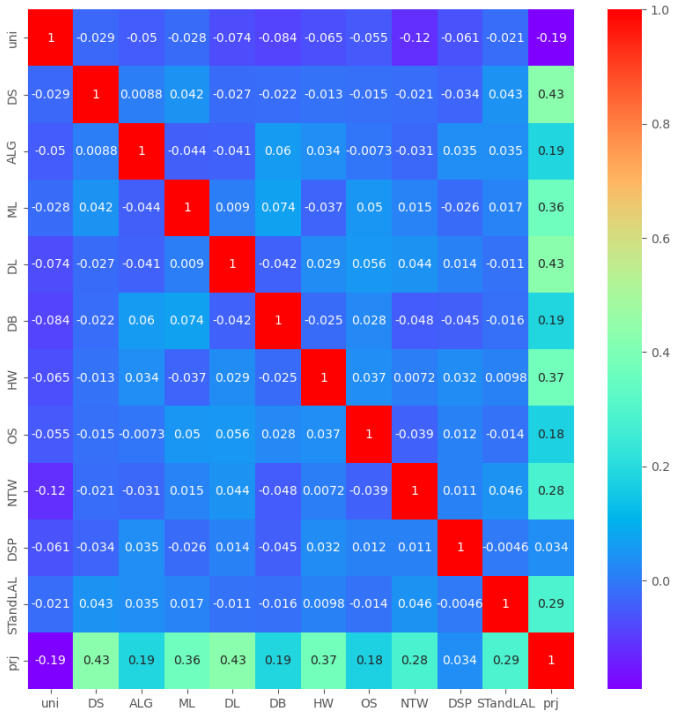


نمودار ۳: توزیع نمرات درس طراحی الگوریتم با میانگین ۱۰.۰۴

به همین ترتیب تمامی نمودارهای توزیع نمرات 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 ۴

و) تقسیم داده‌ها به مجموعه دادهای آموزشی و تستی

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

```
# Check the shape of X_train and X_test

X_train.shape, X_test.shape
```

```
((724, 11), (181, 11))
```

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 می کنیم.

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
```

```
1.0
```

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.

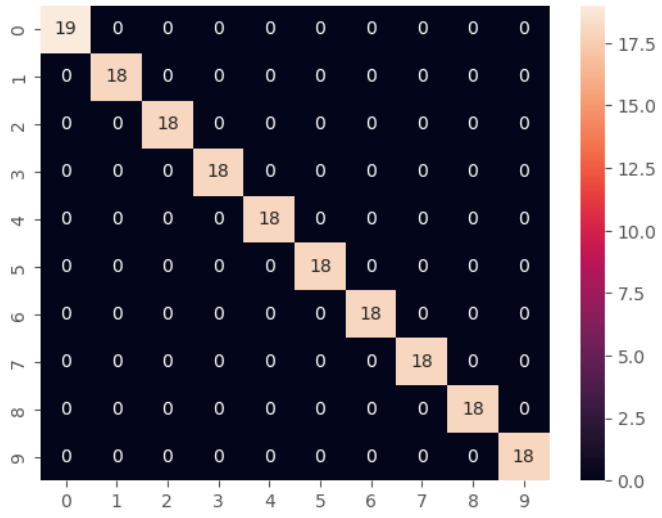
```
# Check for overfitting and underfitting

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:  1.0
score on test data:  1.0
```

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 ۱۰ 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	1.00	1.00	1.00	19
1	1.00	1.00	1.00	18
2	1.00	1.00	1.00	18
3	1.00	1.00	1.00	18
4	1.00	1.00	1.00	18
5	1.00	1.00	1.00	18
6	1.00	1.00	1.00	18
7	1.00	1.00	1.00	18
8	1.00	1.00	1.00	18
9	1.00	1.00	1.00	18
accuracy			1.00	181
macro avg	1.00	1.00	1.00	181
weighted avg	1.00	1.00	1.00	181



نمودار ۵

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 تمامی 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 تمامی پیش‌بینی‌ها را به درستی انجام داده است، مقدار هر سه معیار ۱.۰ است که نتیجه مطلوبی است.

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

```
(1.0, 1.0, 1.0, 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 ۰.۰۰۰۰۰۰۰۰۰۰۰۰۰۴۱ 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: 0.000000000000041
Mean Squared Error: 0.00000000000000000000000002729
Root Mean Squared Error: 0.000000000000052
```

برای آموزش مدل بر پایه MLP از MLPRegressor در sklearn.neural_network استفاده شده است. این ماژول با دریافت تعداد لایه‌های پنهان، تابع فعال‌سازی، حداکثر تعداد تکرارها 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=(270, 230), max_iter = 300, 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 ۰.۹۹ برای مدل به همراه داشت.

```
mlp_reg = MLPRegressor(hidden_layer_sizes=(270, 230), max_iter = 300, 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
```

```
0.990882895523688
```

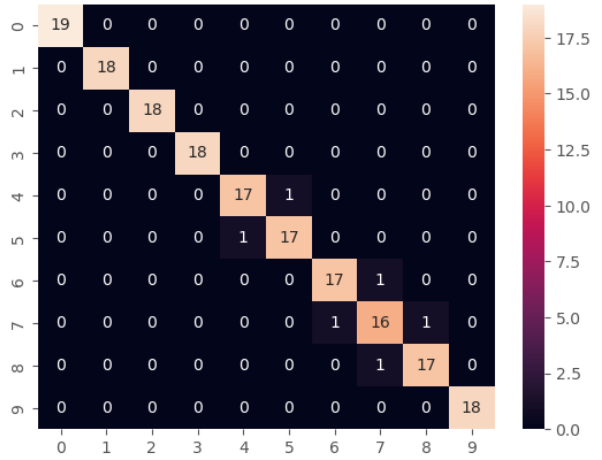
همچنین در ادامه محاسباتی برای تشخیص بیش برازش و کم برازش صورت گرفته است که نشان می‌دهد این دو مشکل رایج در مدل تعلیم دیده با مقادیر فعلی وجود ندارند. برای تشخیص اینکه بیش برازش و یا کم برازش رخ داده است یا خیر، دقت را روی داده‌های

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:  0.9863999256201812
score on test data:  0.990882895523688
```

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 ۱۰ 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	1.00	1.00	1.00	19
1	0.94	0.94	0.94	18
2	0.89	0.89	0.89	18
3	0.94	0.94	0.94	18
4	0.89	0.89	0.89	18
5	0.89	0.89	0.89	18
6	1.00	1.00	1.00	18
7	1.00	1.00	1.00	18
8	0.94	0.94	0.94	18
9	0.94	0.94	0.94	18
accuracy			0.94	181
macro avg	0.94	0.94	0.94	181
weighted avg	0.94	0.94	0.94	181



نمودار ۶

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 تمامی 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 ۰.۹۶ است که نتیجه مطلوبی است.

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

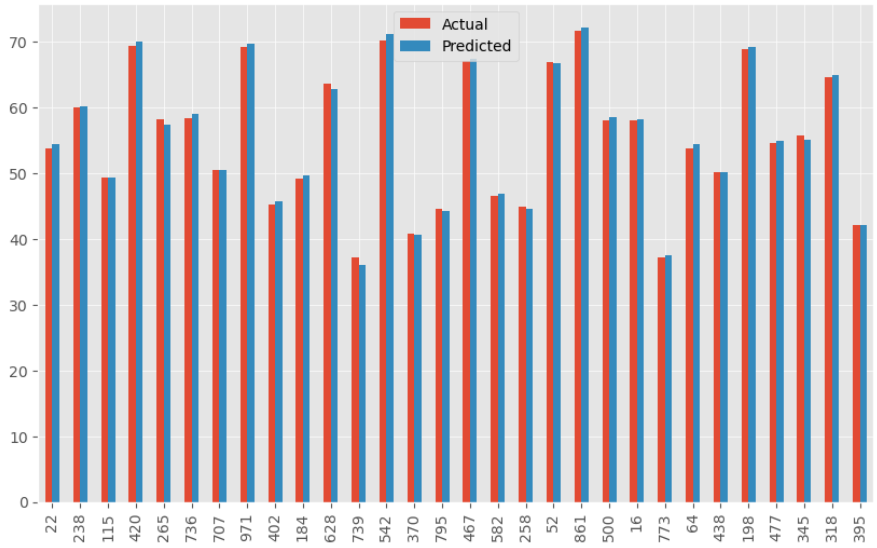
```
(0.9668508287292817, 0.9668508287292817, 0.9668508287292819, None)
```

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 ۰.۵۵ 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.

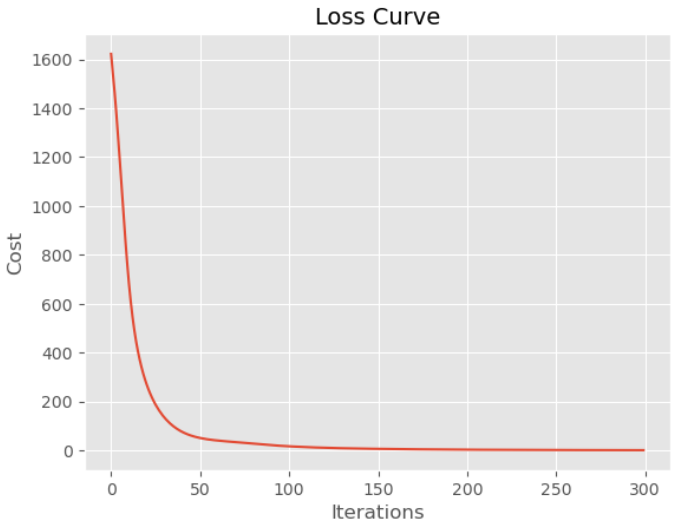
```
Mean Absolute Error: 0.5583226290610795
Mean Squared Error: 1.7575161998562734
Root Mean Squared Error: 1.3257134682337182
```

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.



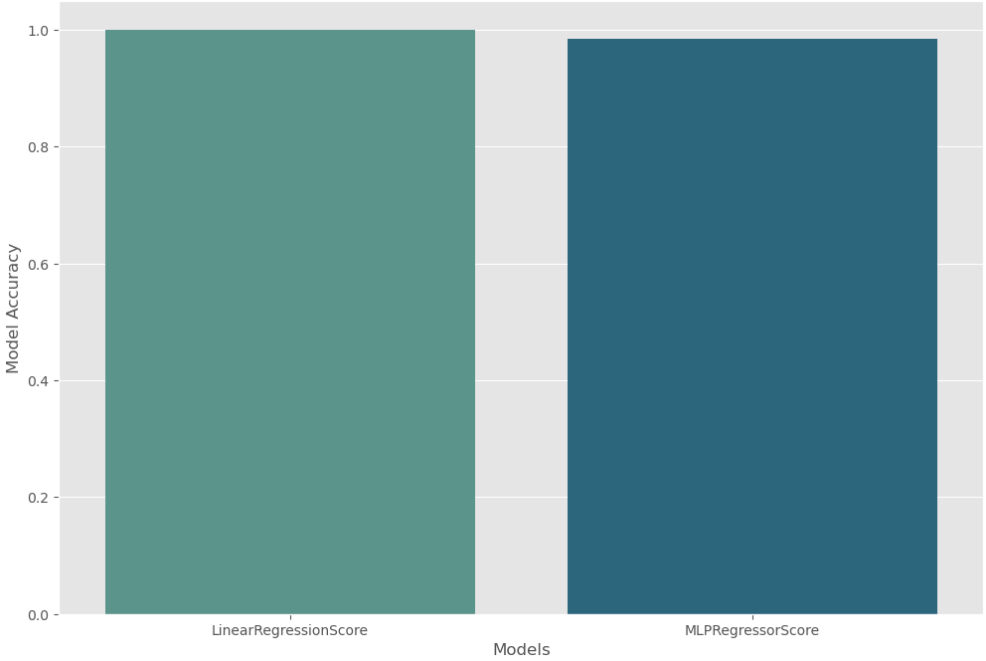
نمودار ۷

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



نمودار ۸

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



نمودار ۹