پاسخ سوال چهارم تمرین دوم هوش مصنوعی

گزارش مراحل تحلیل و پیش‌پردازش داده‌ها و همچنین ارائه مدلی برای تخمین خرید بیمه سفر توسط مشتریان

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.

	Customer Id	Age	Employment Type	GraduateOrNot	AnnualIncome	FamilyMembers	ChronicDiseases	FrequentFlyer	EverTravelledAbroad	TravelInsurance
0	3JUN0VW6F043	34	Private Sector/Self Employed	Yes	1300000	6	0	Yes	No	No
1	VLHY2ABIR4QL	28	Private Sector/Self Employed	Yes	750000	7	0	Yes	No	No
2	6E3F7UNXYNFF	28	Private Sector/Self Employed	Yes	750000	6	0	Yes	No	No
3	JJ8R0ZRYWR31	32	Government Sector	Yes	800000	6	1	No	No	No
4	2WGFUEX6IEHM	34	Private Sector/Self Employed	Yes	700000	4	1	No	No	No

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. طبق خروجی بدست آمده هیچ یک از ویژگی‌ها دارای مقادیر از دست رفته نیستند.

	missing value count
Customer Id	0
Age	0
Employment Type	0
GraduateOrNot	0
AnnualIncome	0
FamilyMembers	0
ChronicDiseases	0
FrequentFlyer	0
EverTravelledAbroad	0
TravelInsurance	0

ج) تمیزسازی داده‌ها

این کار را با حذف ستون شناسه مشتری از داده‌ها آغاز می‌کنیم. چرا که این ویژگی نامرتبط به پردازش‌های آینده تلقی می شود و برای آموزش مدل به آن نیازی نداریم.

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.

	Age	AnnualIncome	FamilyMembers	ChronicDiseases
count	1590.000000	1.590000e+03	1590.000000	1590.000000
mean	29.642138	9.285535e+05	4.753459	0.279874
std	2.914275	3.752353e+05	1.610490	0.449078
min	25.000000	3.000000e+05	2.000000	0.000000
25%	28.000000	6.000000e+05	4.000000	0.000000
50%	29.000000	9.000000e+05	5.000000	0.000000
75%	32.000000	1.250000e+06	6.000000	1.000000
max	35.000000	1.800000e+06	9.000000	1.000000

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'>
RangeIndex: 1590 entries, 0 to 1589
Data columns (total 9 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   Age                  1590 non-null   int64 
 1   Employment Type      1590 non-null   object
 2   GraduateOrNot        1590 non-null   object
 3   AnnualIncome         1590 non-null   int64 
 4   FamilyMembers        1590 non-null   int64 
 5   ChronicDiseases      1590 non-null   int64 
 6   FrequentFlyer        1590 non-null   object
 7   EverTravelledAbroad  1590 non-null   object
 8   TravelInsurance      1590 non-null   object
dtypes: int64(4), object(5)
memory usage: 111.9+ KB
```

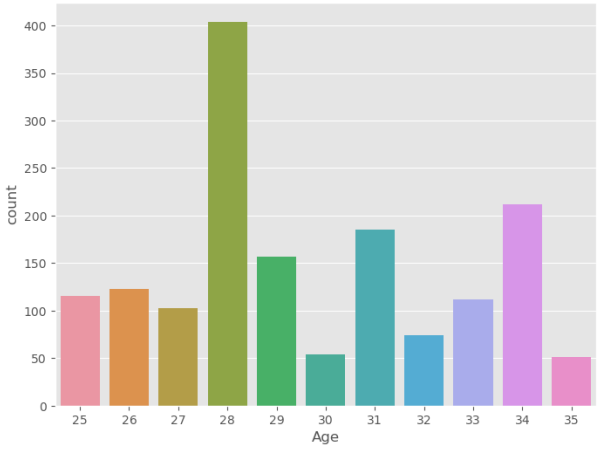
```
df.shape
```

```
(1590, 9)
```

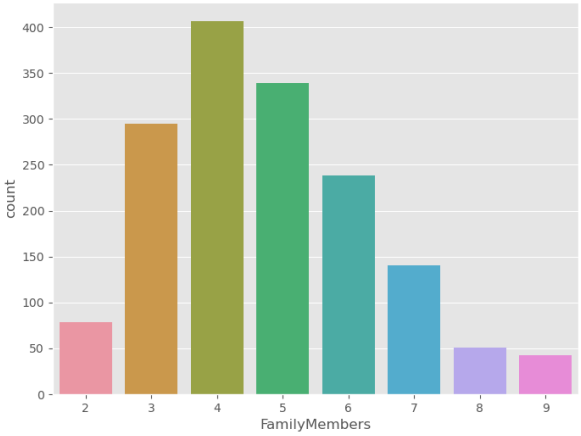
سپس به سراغ رسم نمودارها و data visualization می‌رویم.

نمودار توزیع

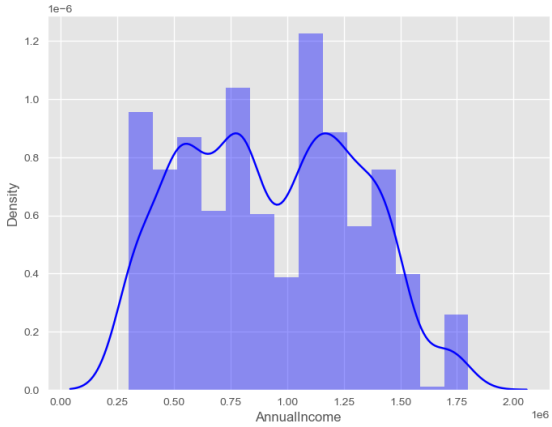
در ادامه با رسم نمودارهایی تعداد مشاهدات در هر دسته از ویژگی‌ها را تصویرسازی می‌کنیم تا دید بهتری نسبت به داده‌ها داشته باشیم.



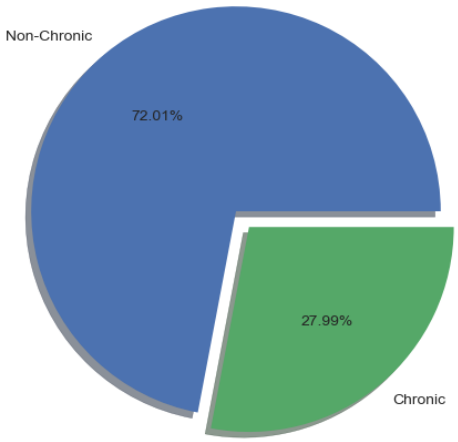
نمودار ۱: سن اکثر مشتریان ۲۸ سال است و افراد با سن ۳۵ سال کمترین تعداد را در میان جامعه مورد بررسی دارند.



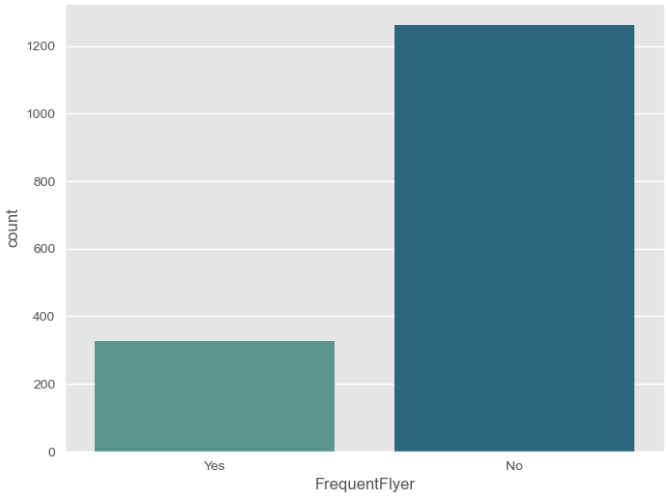
نمودار ۲: بیش‌تر خانوارها در این مجموعه داده ۴ عضو داشته وکمترین فراوانی مربوط به خانواده‌هایی با ۸ و ۹ عضو است.



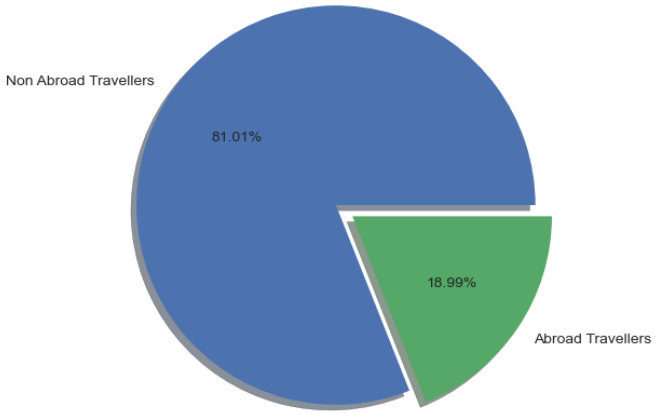
نمودار ۳: شکل بالا نمودار توزیع درآمد سالانه افراد را با میانگین ۹۲۸۵۵۳.۴۵ نشان می‌دهد.



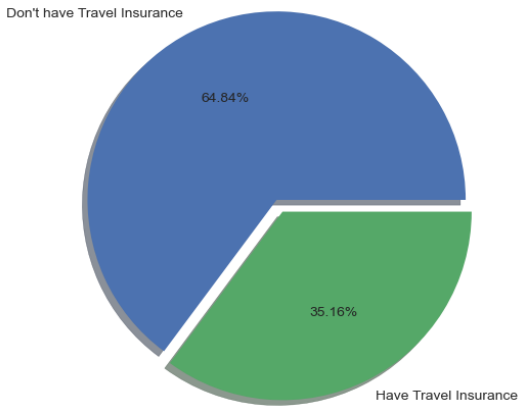
نمودار ۴: اکثر مشتریان از بیماری یا شرایط خاص رنج نمی‌برند.



نمودار ۵: اکثر افراد جز دسته Non Frequent Flyers محسوب می‌شوند.



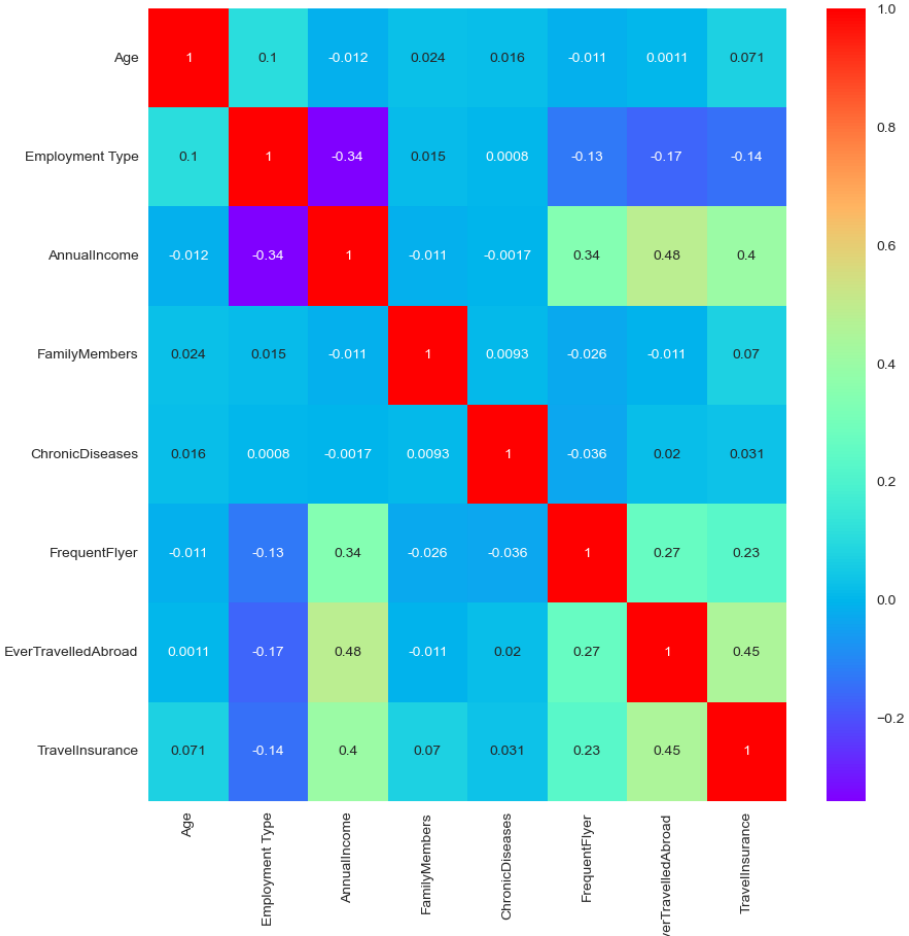
نمودار ۶: اکثر افراد مورد بررسی سفرهای خارجی نداشته‌اند.



نمودار۷: اکثر مشتری‌ها بسته بیمه مسافرتی خریداری نکرده‌اند. از این نمودار می‌توان نتیجه گرفت که مجموعه داده کمی نامتعادل است.

ه) تبدیل داده‌ها به داده‌های عددی و رسم ماتریس هبستگی

در ادامه داده‌های دسته‌بندی شده یا همان categorical را به داده‌های عددی تبدیل کرده و با رسم نمودار هبستگی میزان همبستگی دوبه‌دوی متغیرها به یکدیگر را نمایش می‌دهیم.



نمودار ۸

و) تقسیم داده‌ها به مجموعه دادهای آموزشی و تستی

در این مرحله می‌خواهیم با استفاده از داده‌ها مدلی برای تخمین اینکه بیمه سفر گرفته شده یا نه ارائه کنیم. در نتیجه ویژگی TravelInsurance به عنوان برچسب y و بقیه داده‌ها به عنوان داده‌های ورودی محسوب می‌شوند.

پس 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
```

```
((1272, 8), (318, 8))
```

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 لجستیک و روش‌های بر پایه mlp مدل را آموزش دهیم. ابتدا به آموزش مدل با استفاده از رگرسیون لجستیک می‌پردازیم. با استفاده از ابزار LogisticRegression 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 = LogisticRegression()
lr.fit(X_train, y_train)
y_pred_test = lr.predict(X_test)
LogisticRegressionScore = lr.score(X_test,y_test)
LogisticRegressionScore
```

```
0.779874213836478
```

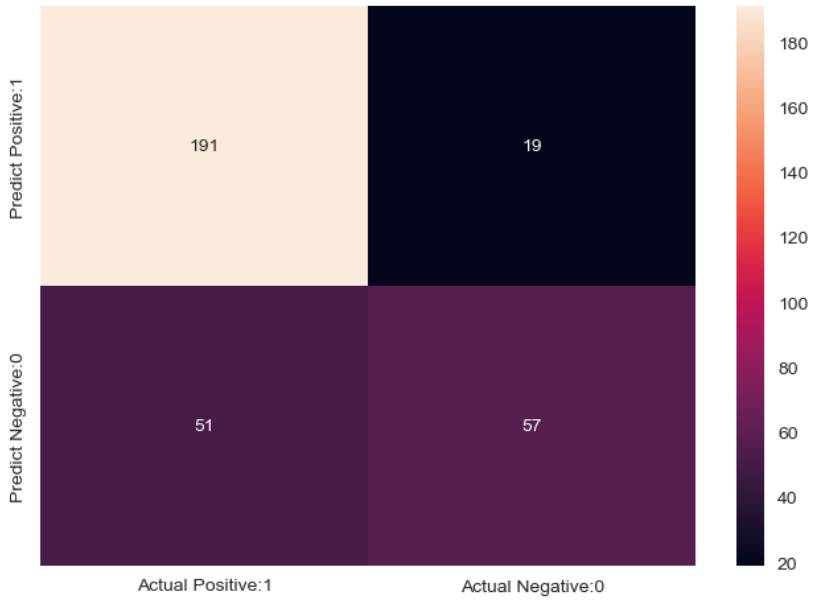
برای تشخیص اینکه بیش برازش و یا کم برازش رخ داده است یا خیر، دقت را روی داده‌های تست و داده‌های آموزشی محاسبه می کنیم. با توجه به آنکه دقت هر دو متناسب با یکدیگر بدست آمده است، نتیجه می گیریم مدل به خوبی آموزش دیده است.

```
# 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:  0.7720125786163522
score on test data:  0.779874213836478
```

در ادامه ماتریس درهم ریختگی ترسیم و معیارهای recall، precision و accuracy محاسبه شده‌اند.



نمودار ۹

	precision	recall	f1-score	support
0	0.79	0.91	0.85	210
1	0.75	0.53	0.62	108
accuracy			0.78	318
macro avg	0.77	0.72	0.73	318
weighted avg	0.78	0.78	0.77	318

ماتریس درهم ریختگی که ابزاری برای اندازه‌گیری کارایی مسائل دسته‌بندی به کمک یادگیری ماشینی است، مقایسه‌ای از مقادیر پیش‌بینی شده و مقادیر واقعی را نشان می‌دهد. مدلی که به خوبی آموزش دیده باشد باید نرخ TP و TN بالایی داشته باشد که با توجه به نمودار بالا، این شرط برآورده شده است (اعداد ۵۷ و ۱۹۱ نشان‌گر این مطلب‌اند.) همچنین یک مدل خوب نرخ FP و FN پایینی دارد که این قید هم در این مدل ارضا شده است (اعداد ۱۹ و ۵۱ این موضوع را نشان می‌دهند.)

از دیگر معیارهای ارزشیابی مدل‌های دسته‌بندی، مقادیر recall، precision و accuracy هستند. صحت در این مدل ۰.۷۷ برآورد شده است که نشان دهنده تعداد دفعات صحیح پیش‌بینی مدل است.

```
classification_accuracy = (TP + TN) / float(TP + TN + FP + FN)
print('accuracy: ', classification_accuracy)
```

```
accuracy:  0.779874213836478
```

دقت یک مدل نشان می‌دهد از بین تمام پیش‌بینی‌هایی که positive در نظر گرفته شده است، چقدر از آن‌ها واقعا positiveاند که در یک مدل خوب باید تا حد امکان درصد بالایی را نشان دهد. در این مدل دقت ۰.۹۰ محاسبه شده است که نتیجه مطلوبی‌ست.

```
precision = TP / float(TP + FP)
print('precision: ', precision)
```

```
precision:  0.9095238095238095
```

در نهایت معیار recall با مقدار ۰.۷۸ نسبت داده‌هایی که در گروه positive دسته‌بندی شده‌اند را به تعداد کل کلاس‌های positive نشان می‌دهد.

```
recall = TP / float(TP + FN)
print('recall: ', recall)
```

```
recall:  0.7892561983471075
```

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 ۰.۲۲ 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: 0.22012578616352202
Mean Squared Error: 0.22012578616352202
Root Mean Squared Error: 0.46917564532222045
```

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

(random_state = 98, hidden_layer_sizes = (10, 5), max_iter = 300)

این اعداد بر حسب حدس اولیه تنظیم شده‌اند و به دلیل آنکه در آزمون و خطای پس از آن مشخص شد که دقت مدل با استفاده از این پارامترها بهترین نتیجه قابل دریافت را بدست می‌آورد، نیازی به تغییر آن احساس نشد. 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 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:  0.8207547169811321
score on test data:  0.8176100628930818
```

در ادامه ماتریس درهم ریختگی ترسیم و معیارهای recall، precision و accuracy محاسبه شده‌اند.

نمودار ۱۰

	precision	recall	f1-score	support
0	0.83	0.90	0.87	210
1	0.78	0.65	0.71	108
accuracy			0.82	318
macro avg	0.81	0.78	0.79	318
weighted avg	0.81	0.82	0.81	318

ماتریس درهم ریختگی که ابزاری برای اندازه‌گیری کارایی مسائل دسته‌بندی به کمک یادگیری ماشینی است، مقایسه‌ای از مقادیر پیش‌بینی شده و مقادیر واقعی را نشان می‌دهد. مدلی که به خوبی آموزش دیده باشد باید نرخ TP و TN بالایی داشته باشد که با توجه به نمودار بالا، این شرط برآورده شده است (اعداد ۷۰ و ۱۹۰ نشان‌گر این مطلب‌اند.) همچنین یک مدل خوب نرخ FP و FN پایینی دارد که این قید هم در این مدل ارضا شده است (اعداد ۲۰ و ۳۸ این موضوع را نشان می‌دهند.)

از دیگر معیارهای ارزشیابی مدل‌های دسته‌بندی، مقادیر recall، precision و accuracy هستند. صحت در این مدل ۰.۸۱ برآورد شده است که نشان دهنده تعداد دفعات صحیح پیش‌بینی مدل است.

```
classification_accuracy = (TP + TN) / float(TP + TN + FP + FN)
print('accuracy: ', classification_accuracy)
```

```
accuracy:  0.8176100628930818
```

دقت یک مدل نشان می‌دهد از بین تمام پیش‌بینی‌هایی که positive در نظر گرفته شده است، چقدر از آن‌ها واقعا positiveاند که در یک مدل خوب باید تا حد امکان درصد بالایی را نشان دهد. در این مدل دقت ۰.۹۰ محاسبه شده است که نتیجه مطلوبی‌ست.

```
precision = TP / float(TP + FP)
print('precision: ', precision)
```

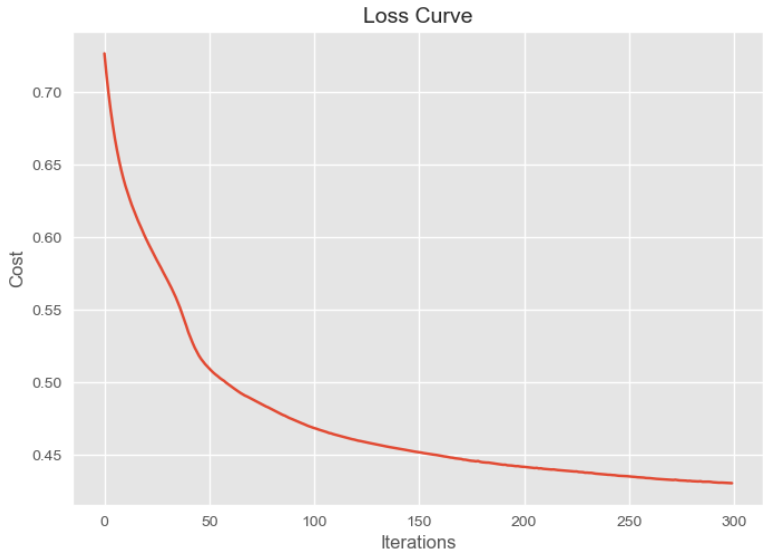
```
precision:  0.9047619047619048
```

در نهایت معیار recall با مقدار ۰.۸۳ نسبت داده‌هایی که در گروه positive دسته‌بندی شده‌اند را به تعداد کل کلاس‌های positive نشان می‌دهد.

```
recall = TP / float(TP + FN)
print('recall: ', recall)
```

```
recall:  0.8333333333333334
```

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



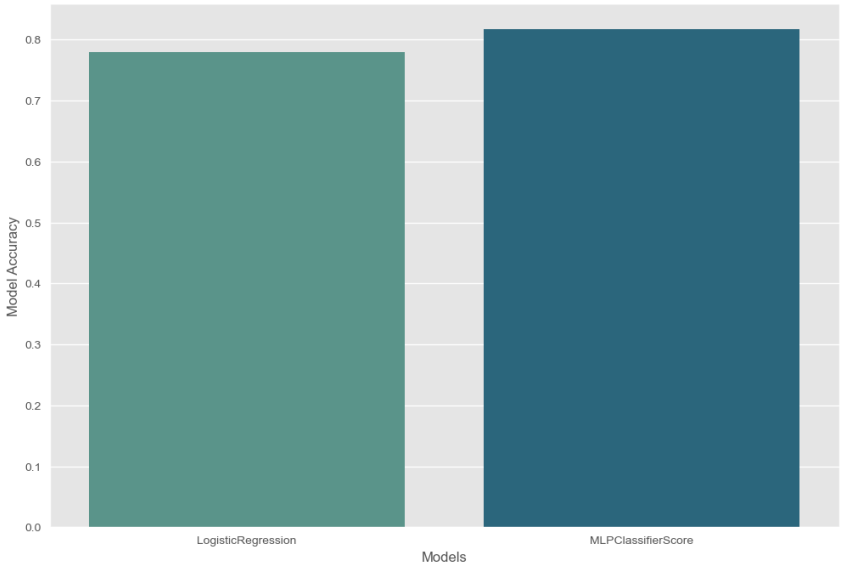
نمودار ۱۱

برای اطمینان از عملکرد بهینه مدل اما مقادیر مختلفی از hidden_layer_sizes و activation در یک حلقه مورد استفاده قرار گرفته و مدل‌های مختلفی با استفاده از پارامترهای گوناگون آموزش داده می‌شود و نتیجه دقت مدل آموزش داده شده با پارامترهای متنوع بر روی داده‌های تست و آموزشی محاسبه می شود.

```
(tanh) neuron:(20, 15), accuracy_test:0.8018867924528302, accuracy_train:0.8474842767295597
(tanh) neuron:(15, 10), accuracy_test:0.8176100628930818, accuracy_train:0.8419811320754716
(tanh) neuron:(12, 8), accuracy_test:0.8113207547169812, accuracy_train:0.835691823899371
(tanh) neuron:(10, 5), accuracy_test:0.8113207547169812, accuracy_train:0.8333333333333334
(relu) neuron:(20, 15), accuracy_test:0.8081761006289309, accuracy_train:0.8553459119496856
(relu) neuron:(15, 10), accuracy_test:0.8113207547169812, accuracy_train:0.8372641509433962
(relu) neuron:(12, 8), accuracy_test:0.779874213836478, accuracy_train:0.835691823899371
(relu) neuron:(10, 5), accuracy_test:0.7924528301886793, accuracy_train:0.8238993710691824
```

همانطور که از نتایج برمی‌آید، افزایش پیچیدگی مدل با تابع فعال‌سازی tanh مدل را به سمت بیش برازش می‌برد. همچنین با استفاده از تابع فعال‌سازی relu فاصله عملکرد مدل در داده‌های تست و آموزشی، نسبت به نتایج در شبکه با تابع فعال‌سازی tanh، بیش‌تر بوده که نشان دهنده حرکت مدل به سمت overfitting است.

در نهایت می‌توان نتیجه آموزش با رگرسیون لجستیک و دسته‌بندی بر پایه MLP را در نمودار زیر نشاهده کرد.



نمودار ۱۲