به نام خدا

دانشگاه صنعتی شریف

دانشکده مهندسی برق

مقدمه ای بر یادگیری ماشین ـ دکتر صالح کلیبر

نیم‌سال اول ۱۳۹۹

گزارش بخش کامپیوتری تمرین سری دوم

امید شرفی (۹۶۱۰۱۸۳۸)

---

۱ سوال یک

در ابتدا دیتافریم را از ورودی csv لود میکنیم.

```
# load dataset
data = pd.read_csv("JuiceQuality.csv")
data = data.loc[:, 'fixed acidity':'quality']
display(data.head(5))

# X=
X = data.loc[:, 'fixed acidity':'Vit Indx']

# y=
Y = data['quality']
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	A Indx	density	pH	sulphates	Vit Indx	quality
0	5.9	0.32	0.39	low	0.114	24.0	140.0	13.905151	0.99340	3.09	low	very_low	high
1	7.8	0.24	0.18	high	0.046	33.0	160.0	17.130070	0.99630	3.20	medium	low	high
2	7.7	0.58	0.01	very_low	0.088	12.0	18.0	15.618966	0.99568	NaN	medium	low	high
3	8.3	0.18	0.30	very_low	0.033	20.0	57.0	18.085322	0.99109	3.02	medium	medium	high
4	6.5	NaN	0.31	very_low	0.044	NaN	127.0	13.968160	0.99280	3.49	low	low	high

شکل ۱: داده های ورودی

در ادامه داده های مربوط به بردار ورودی ویژگی ها و همچنین بردار خروجی که هردو فرمت غیر عددی دارند را با معادل های عددی مناسب در ویژگی ها و لیبل های باینری برای خروجی جایگزین کرده و سپس NaN ها را نیز به دو روش ذکر شده در سوال مدیریت میکنیم.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	A Indx	density	pH	sulphates	Vit Indx
0	5.9	0.32	0.39	2.0	0.114	24.0	140.0	13.905151	0.99340	3.09	1.0	1.0
1	7.8	0.24	0.18	4.0	0.046	33.0	160.0	17.130070	0.99630	3.20	2.0	2.0
2	7.7	0.58	0.01	1.0	0.088	12.0	18.0	15.618966	0.99568	NaN	2.0	2.0
3	8.3	0.18	0.30	1.0	0.033	20.0	57.0	18.085322	0.99109	3.02	2.0	3.0
4	6.5	NaN	0.31	1.0	0.044	NaN	127.0	13.968160	0.99280	3.49	1.0	2.0
...	...	...	...	...	...	...	...	...	...	...	...	...
6492	6.0	0.17	0.33	3.0	0.036	30.0	111.0	13.602140	0.99362	3.32	2.0	2.0
6493	7.0	0.31	0.31	5.0	0.036	45.0	NaN	15.227424	NaN	2.98	1.0	3.0
6494	7.3	0.26	0.33	5.0	NaN	48.0	127.0	16.240930	0.99693	NaN	2.0	2.0
6495	6.4	0.23	0.35	3.0	0.039	43.0	147.0	14.171199	0.99216	3.18	1.0	3.0
6496	6.3	0.30	0.29	2.0	0.048	33.0	142.0	13.925044	0.98956	3.22	1.0	4.0

6497 rows × 12 columns

	quality
0	1.0
1	1.0
2	1.0
3	1.0
4	1.0
...	...
6492	1.0
6493	1.0
6494	1.0
6495	1.0
6496	1.0

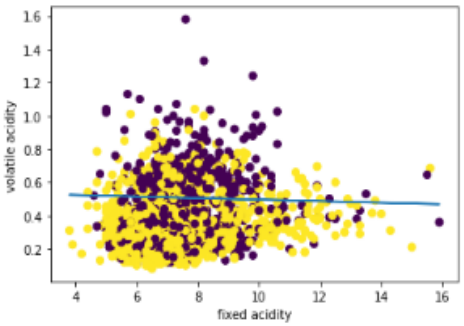
6497 rows × 1 columns

شکل ۲: داده های عددی شده(پیش از جایگذینی یا حذف NaN ها)

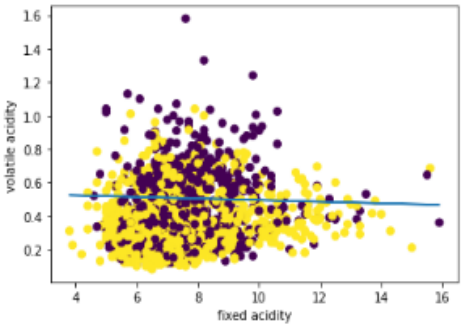
۲ سوال دو

در این مساله در ابتدا کلاس مربوط به پیاده سازی طبقه بند LogisticRegression را آماده میکنیم. همچنین با توجه به آن که تابع MSE برای تابع sigmoid یک تابع محدب محسوب نمیشود از تابع Cross-Entropy استفاده میکنیم که روش های بهینه سازی محدب ما درست پاسخ دهند.

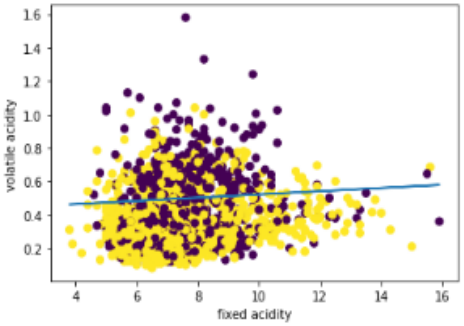
```
Model accuracy for removed NaN method and LR =  0.01  : 66.31923764145324
```



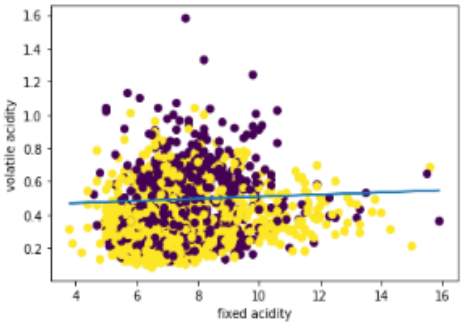

```
Model accuracy for removed NaN method and LR =  0.1  : 66.25967837998809
```




```
Model accuracy for removed NaN method and LR =  1  : 66.49791542584872
```

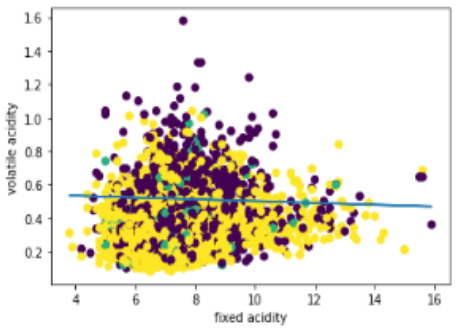



```
Model accuracy for removed NaN method and LR =  10  : 66.43835616438356
```

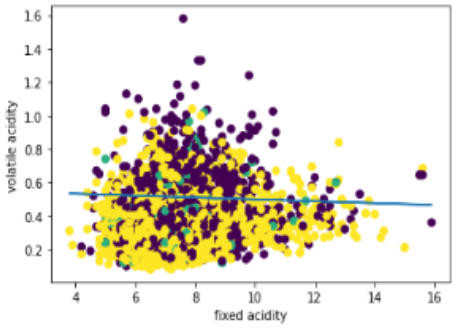


شکل ۳: نتایج بدون حذف داده های پرت و با روش حذف داده های گم شده

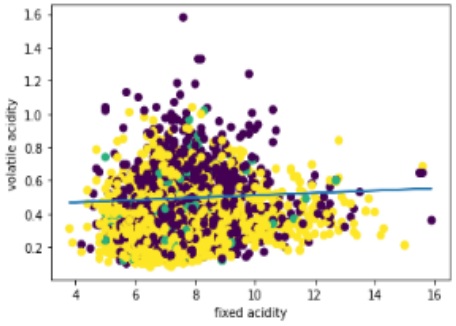
```
Model accuracy for replaced NaN method and LR =  0.01  : 62.92134831460674
```



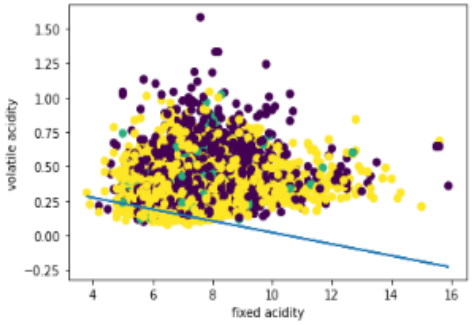

```
Model accuracy for replaced NaN method and LR =  0.1  : 62.92134831460674
```




```
Model accuracy for replaced NaN method and LR =  1  : 62.70586424503617
```

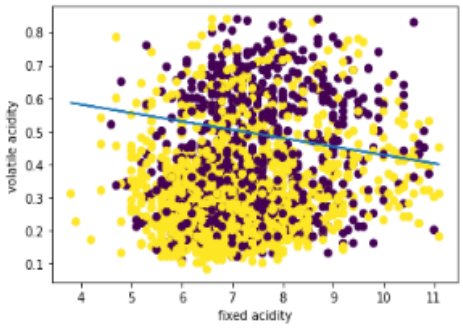



```
Model accuracy for replaced NaN method and LR =  10  : 38.340772664306606
```

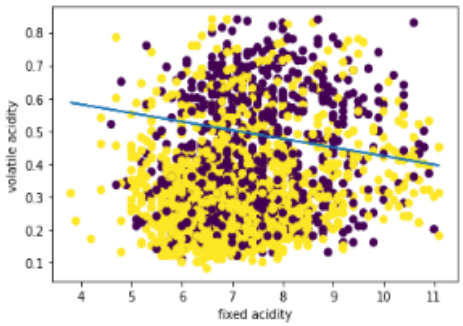


شکل ۴: نتایج بدون حذف داده های پرت و با روش جایگذینی داده های گم شده

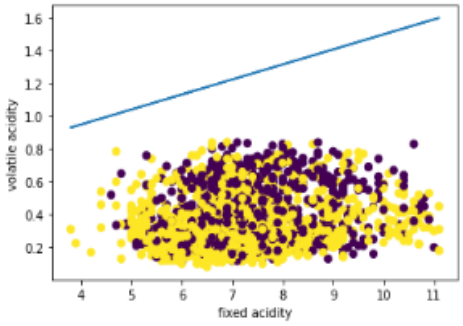
```
Model accuracy for removed NaN method and LR =  0.01  : 66.49166151945646
```



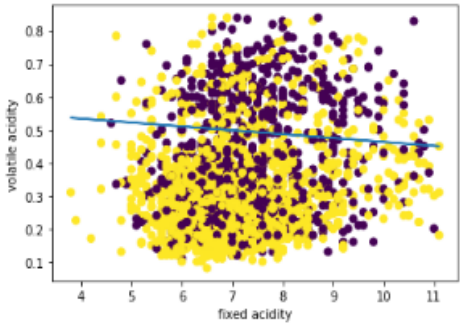

```
Model accuracy for removed NaN method and LR =  0.1  : 66.52254478072884
```




```
Model accuracy for removed NaN method and LR =  1  : 63.650401482396546
```

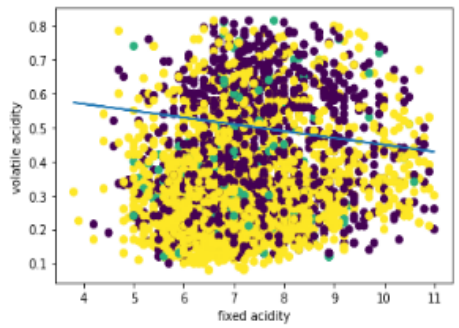



```
Model accuracy for removed NaN method and LR =  10  : 66.21371216800495
```

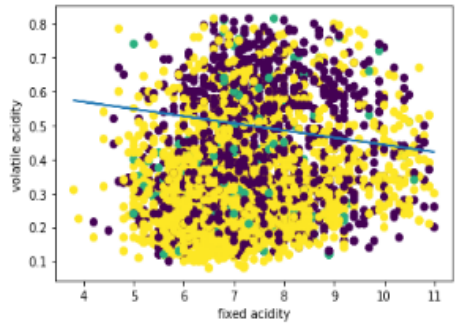


شکل ۵: نتایج با حذف داده های پرت و با روش حذف داده های گم شده

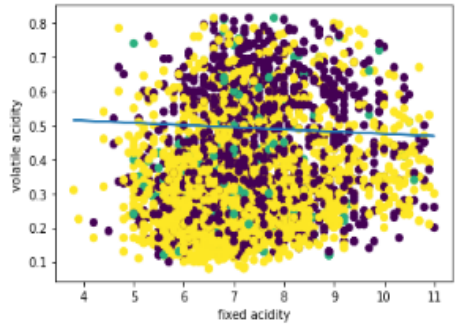
```
Model accuracy for replaced NaN method and LR =  0.01  : 62.67347264316477
```



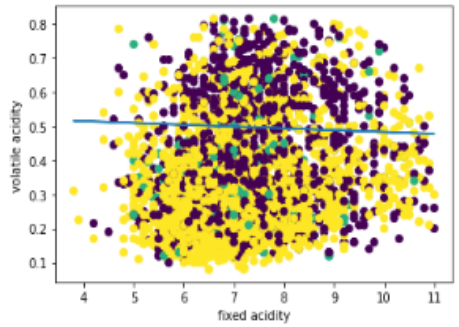

```
Model accuracy for replaced NaN method and LR =  0.1  : 62.80108470250438
```




```
Model accuracy for replaced NaN method and LR =  1  : 62.56181209124262
```




```
Model accuracy for replaced NaN method and LR =  10  : 62.52990907640772
```



شکل ۶: نتایج با حذف داده های پرت و با روش جایگذینی داده های گم شده

با توجه به نمودارهای بالا در وحله ی اول این نکته مهم است که در حقیقت داده ها در این دو بعد ویژگی

یادگیری پذیر نبودن چرا که دقت ما بسیار پایین و در حدود شصد درصد میباشد. همچنین در نتیجه با توجه به این دیتاست شاید نتیجه گیری در مورد پارامترها و تاثیر آن ها بر روی فرآیند یادگیری در این شرایط خیلی معتبر نباشد، با این حال نتایج زیر قابل بحث و بررسی هستند.

- در ابتدا در مورد تاثیر ضریب آموزش، به صورت کلی با کوچکتر بودن این عدد عموما دقت مدل حاصله بالاتر بوده کما این که در شکل نخست دقت مدل با افزایش این عدد تا حدودی کاهش یافته است.
- در مورد این که حذف کردن داده های با مقدار گم شده یا جایگذین کردن آن ها نتیجه ی بهتری در اختیار ما قرار میدهند به صورت کلی باید توجه کرد که علت گم شدن آن ویژگی ها چیست. به طور مثال فرض کنید یکی از ویژگی ها برای قسمت خاصی از داده حذف شده باشد. مثلا یک منطقه از ایران در گزارش بیماران خود یک ویژگی را اندازه گیری نکرده یا در سیستم و دیتاهای خود گزارش نکنند. خب در این شرایط وقتی ما بخواهیم کل این داده ها را حذف کنیم اشتباه بزرگی مرتکب شده ایم و داریم قسمت قابل توجهی از داده را مشاهده نمیکنیم. در عین حال نتایج ما در این جا نشان میدهد که خروجی های مدل برای شرایطی که این داده ها را حذف میکنیم بالاتر بوده. علت مشخص است، چرا که ما داریم در مدلی که داده های گم شده را جایگذین میکنیم، در حقیقت انگار حجمی از داده های تا حدودی ناصحیح و خطا دار وارد مدل میکنیم، در نتیجه طبیعتا این مدل بر روی داده های آموزش دقت پایین تری خواهند داشت اما در شرایطی که بخواهیم دقت را بر اساس داده های اعتبارسنجی ارزیابی کنیم ممکن است نتایج با توجه به نکته ی ذکر شده در مدل های جایگذین کننده شرایط بهتری داشته باشید.
- در نهایت در مورد تاثیر حذف داده های پرت به صورت کلی حذف داد های پریت عموما تاثیر مثبتی میتواند داشته باشید. البته این که معیار پرت بودن را متناسب با توزیع و حجم داده ها درست انتخاب کنیم تاثیرگذار خواهد بود. حقیقتا در این دیتاست با توجه به خطای بالایی که داریم خیلی نتیجه گیری در مورد تاثیر حذف داده های پرت بر اساس نتایجی که ما در اینجا به دست آورده ایم خیلی مشهود نمیباشد.

۳ سوال سه

۱.۳ الف

با استفاده از جدولی که شامل داده های جایگذین شده با حذف NaN ها در بخش قبل بوده کار کرده و صرفا ستون های مربوطه و ۳۷۰ عنصر اول آن را جدا و X و Y موردنیاز این مساله را شکل میدهیم.

```
X = data_nan_drop[['fixed acidity', 'volatile acidity', 'citric acid']][:370].values
Y = data_nan_drop[['A Indx']][:370].values
```

```
numpy.ndarray
```

شکل ۷: آماده سازی بردارهای ورودی و خروجی

۲.۳ ب و ج

پس از پیاده سازی الگوریتم با قرار دادن $\eta = ۰/۰۱$ نتیجه ی زیر برای تعداد مراحل زیر حاصل شد.

```
X_train = X[:300, :]
Y_train = Y[:300]

# Gradient Descent Algorithm setting
ETA = 0.01
W = [5, -5, 5, -5]
max_iteration = 100000
min_error = 0.095

W, MSE, i = GD(X_train, Y_train, ETA, W, max_iteration, min_error)
print('W =', W, '\n')
print('MSE =', MSE, '\n')
print('GD iteration number  =', i)
```

```
W = [ 2.08241273  0.45489345  2.90072116 -0.07537464]

MSE = 0.095

GD iteration number  = 5568
```

شکل ۸: خروجی برای مقدار بهینه ی ۰.۰۹۵

۳.۳ د

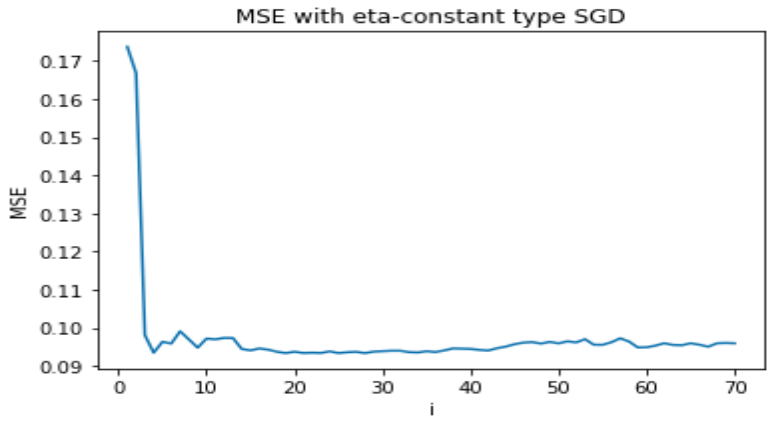
مقدار r با توزیع یکنواخت بین ۱ تا m انتخاب میکنیم. در نتیجه احتمال انتخاب هرکدام $\frac{۱}{m}$ بوده و در نتیجه داریم :

$$E[V_t|X^{(t)}] = \frac{۱}{m}\sum_{i=۱}^{n}\nabla l_i((W,b)^{(t)}) = \nabla L(X^{(t)})$$

۴.۳ ه

با همان مقدار $\eta = ۰/۰۱$ نمودار MSE برحسب تعداد داده های دیده شده پس از سیصد داده ی نخست و با فرض شروع از بردار وزن به دست آماده در بخش قبل به صورت زیر به دست آمد.

```
W = [ 2.09095801  0.45532368  2.90082895 -0.07521237]
MSE(70) = 0.0959202343242859
Learning rate = 0.01
```



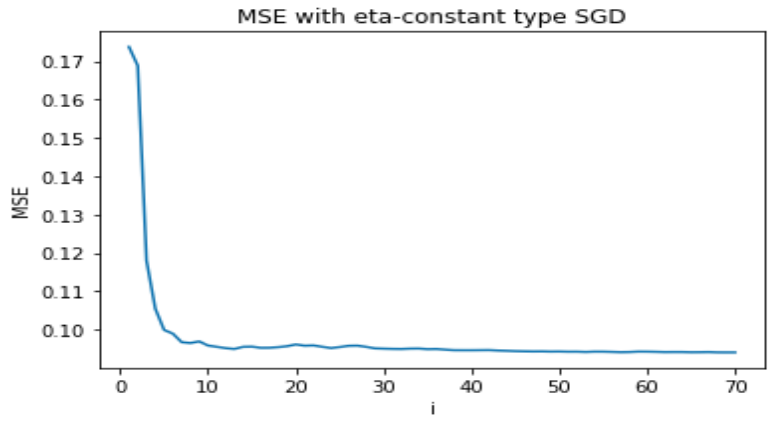
شکل ۹: نمودار MSE در روش یادگیری نرخ ثابت

همانطوری که مشاهده میشود MSE نمودار و مقادیر قابل قبولی دارد. در داده های انتهایی به نظر میرسد که دوباره نمودار اندکی رفتار افزایشی پیدا کرده است اما با توجه به آن که در کل مقدار MSE چه در نقطه ی ۷۰ و چه در سیر کلی نمودار رفتار قابل قبولی دارد.

۵.۳ و

مشابه بخش قبل برای روش یادگیری با نرخ کاهشی نتایج به صورت زیر است. همانطوری که میدانیم چون نرخ یادگیری کاهش است پاسخ نسبت به روش بخش قبل نوسان کمتری دارد.

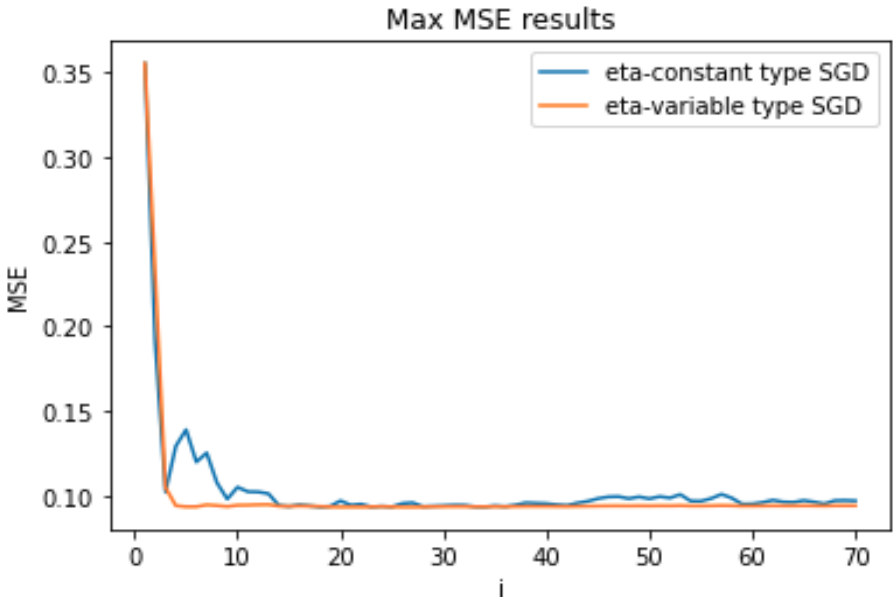
```
W = [ 2.08063779  0.45450703  2.90013622 -0.07659323]
MSE(70) = 0.09403359618397074
Learning rate = 0.01
```



شکل ۱۰: نمودار MSE در روش یادگیری نرخ متغیر

۶.۳ ز

برای بخش آخر با توجه به آن که نوسان در روش اول با نرخ ثابت بیشتر بود احتمالا روش نرخ ثابت ما را در انتخاب حداکثر نرخ آموزش محدود خواهد کرد.



شکل ۱۱: نمودار MSE دو روش در شرایط همگرایی

همانطور که مشاهده میکنید پیش بینی ما درست بود و حداکثر ضریب همگرا $\eta_* = ۰/۰۱۸۴$ میباشد. توجه شود که شرط $MSE^{(۷۰)} <= ۰/۱۵$ ما را محدود نکرد و در هر دو روش همانطور که مشاهده میشود در هر دو روش این اعداد در حدود ۰/۱ میباشد. در حقیقت چیزی که ما را در همگرایی یادگیری محدود کرد، بخش اول الگوریتم روی سیصد داده اول است. وگرنه اگر بخواهیم روی بخش آخر الگوریتم در نظر بگیریم میتوانیم نرخ یادگیری بالاتری هم در نظر بگیریم که با تعریف سوال همگرا باشد.