

باسمه تعالی

گزارش کار تمرین اول درس یادگیری ماشین – جناب آقای دکتر باباعلی

ایمان کیانیان

۱. مقدمه

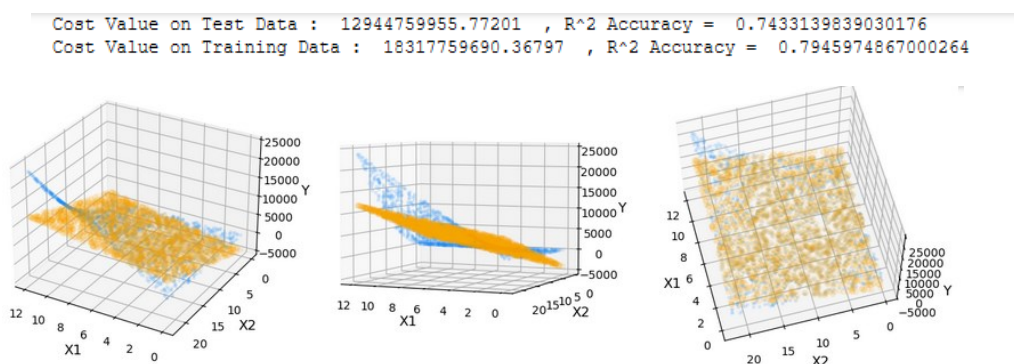
در این گزارش به تهیه نتایج مشاهده شده در این تمرین می پردازیم. البته شکل ها و کد ها در فایل `ipynb` موجود است اما در این فایل می خواهیم گزارشی از خلاصه نتایج و مشاهدات بیاوریم. در ابتدا داده ها توسط پکیج `numpy` لود شد. این داده شامل ۸۰۰۰ نمونه آموزشی شامل دو فیچر و ۲۰۰۰ داده ی تست است که از تابعی درجه سه که در `pdf` سوال آمده پیروی میکنند. بدون توجه به داده های تست و بدون پیش فرضی از تابع هدف در ابتدا کد هایی برای `SSE` و گرادیان نزولی و `Closed-Form` نوشته شد که هر بخش وظایف خود را انجام می دهند. تابعی به نام `Regression` نوشتیم که ورودی هایی همچون n (درجه رگرسیون چند جمله ای — درجه ۱ یعنی رگرسیون خطی) و X و Y و `iteration` و `threshold` و `method` (که روش انتخابی ما `Gradient Descent` است یا `Closed Form`). توابع مورد نیاز را در خود فراخوانی میکند از جمله پیش پردازش داده ها (ساختن داده های آموزشی با توجه به درجه چند جمله ای و داده های خام X) و همچنین `Gradient Descent` یا `Closed` را بسته به مقدار ورودی `method` فراخوانی کرده و مقدار W را پیدا میکند. در قسمت اول رگرسیون خطی، درجه ۳ و ۵ را توسط فرم بسته (نرمال) و گرادیان نزولی را پیاده سازی کردیم. سپس با استفاده از `5 fold cross validation` ، مقادیر مختلف ذکر شده را برای پارامتر منظم سازی بررسی کرده و گزارشی را تهیه کرده ایم که در ادامه بررسی خواهیم کرد.

۲. رگرسیون خطی

در این قسمت به سادگی میتوانیم پیاده سازی لازم را انجام دهیم که در فایل کد آورده شده است. چون در این قسمت بایاس زیاد است پس نمیتوانیم کاری انجام دهیم که دقت را نزدیک به ۱۰۰ درصد کنیم. مدل واقعی درجه ۳ است ولی ما داریم درجه ۱ فیت میکنیم.

• رگرسیون خطی با معادله نرمال

در این قسمت رگرسیون خطی با معادله نرمال پیاده سازی شد. مقدار خطا روی داده های تست و آموزشی تقریباً زیاد است. علت این است که معادله درجه ۱ توان فیت کردن داده های با توزیع چند جمله ای درجه ۳ را ندارد عملاً یک خط نمیتواند یک منحنی را فیت کند. پس باید درجه را بیشتر کنیم. توجه داشته باشیم در معادله نرمال چون ، مقدار W بهینه به انتخاب اولیه W و نرخ یادگیری و تعداد تکرار بستگی ندارد، حداکثر دقت و هزینه کمتری را دریافت میکنیم. اطلاعات بدست آمده در این مرحله :

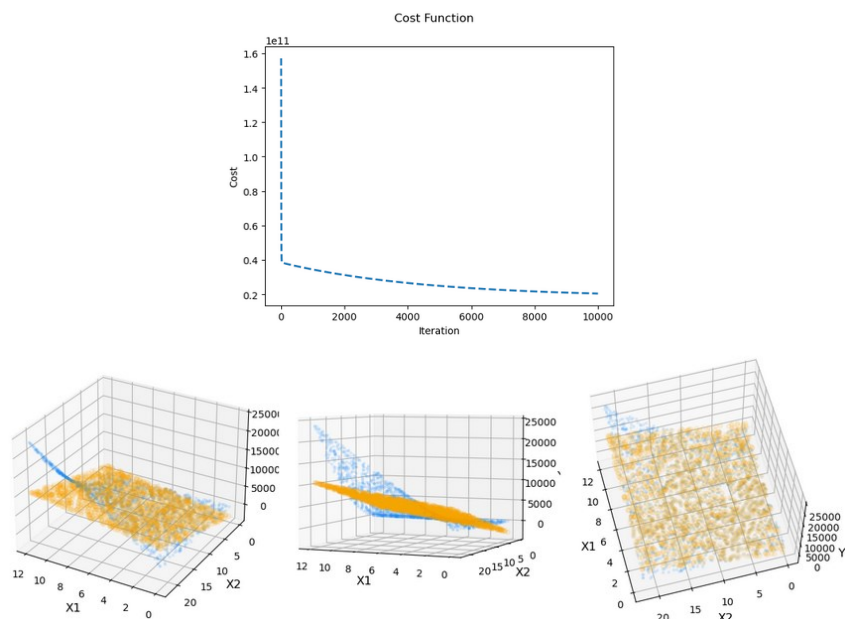


• رگرسیون خطی با گرادیان نزولی

در این قسمت دقت تقریباً مشابه با معادله نرمال داریم و اگر دقت کمی کمتر است به علت تعداد تکرار های غیر بهینه ، انتخاب غلط اولیه ی پارامتر ها ، نرخ یادگیری غیر بهینه و ... است که با استفاده از `GridySearch` میتوانیم در آینده این مشکل را حل کنیم. هر چه نرخ یادگیری بیشتر باشد به شرطی که واگرایی رخ ندهد همگرایی بهتر و سریعتری

خواهیم داشت و هر چه نرخ یادگیری کوچکتر باشد ، یادگیری کند خواهد بود و ممکن است به شرایط ایده آل برای وزن ها نرسیم. نتایج به شرح زیر است که نشان میدهد مدل ما به علت شرایط ذکر شده آموزش کاملی دیده نشده است. این نتایج با ۱۰۰۰۰ تکرار و نرخ یادگیری 10^{-7} حاصل شده است:

Cost Value on Test Data : 15894720003.55463 , R² Accuracy = 0.6848182300305833
 Cost Value on Training Data : 20527575718.899216 , R² Accuracy = 0.7698181592132944



۳. رگرسیون درجه سه

در این قسمت انتظار داریم که دقت بسیار بالایی در هر دو روش معادله نرمال و گرادیان نزولی مشاهده کنیم چون تمامی روابط بین پارامترها در نظر گرفته شده است و تابع هدف ما نیز درجه ۳ است. پس باید حداکثر دقت را مشاهده نماییم.

• رگرسیون درجه ۳ با معادله نرمال (بسته)

در این قسمت به بررسی نتایج حاصل شده توسط پیاده سازی با شرایط یکسان با رگرسیون خطی نرمال میپردازیم. در این نتایج مشاهده می شود دقت ۱۰۰٪ و خطای بسیار بسیار کم بدست آورده ایم. توجه کنیم چون در روش معادله نرمال هیچ هایپر پارامتری نداریم ، بنابراین حداکثر دقت را خواهیم گرفت :

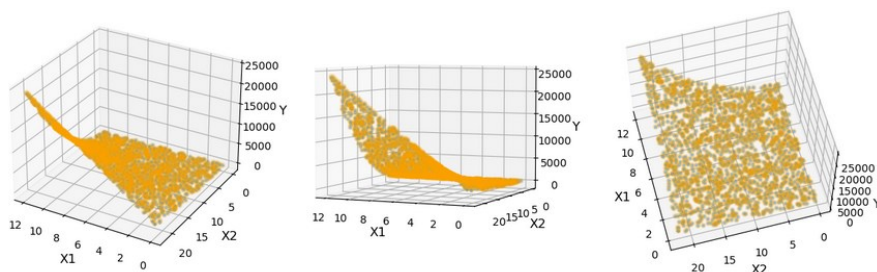
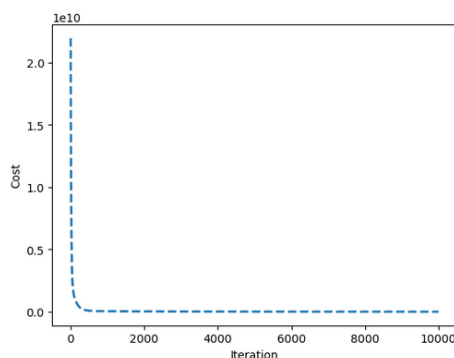
Cost Value on Test Data : 3.0963698027565935e-15 , R² Accuracy = 1.0
 Cost Value on Training Data : 8.665589452395922e-15 , R² Accuracy = 1.0

• رگرسیون درجه ۳ با گرادیان نزولی :

با توجه به دلایلی که در بالا هم ذکر شد، دقت بالایی برای معادله درجه ۳ خواهیم گرفت اما همچنان از وضعیتی که از معادله نرمال استفاده کردیم دقت کمتری داریم که علت آن را بررسی کردیم. علت این است که باید هایپر پارامترهایی از جمله W اولیه ، تعداد تکرار و نرخ یادگیری را مشخص کنیم که ممکن است به بهینه سراسری نرسیم. اول نرخ

یادگیری را 10^{-7} در نظر گرفتیم و عدم همگرایی رخ داد. مقادیر نرخ یادگیری را کوچک و کوچکتر کردیم و در پایان با در نظر گرفتن تعداد تکرار ۱۰۰۰۰ و نرخ یادگیری 10^{-11} نتایج زیر حاصل شده است:

```
Cost Value on Test Data : 3966555.3221134352 , R^2 Accuracy = 0.9999213458351688
Cost Value on Training Data : 6293647.008267477 , R^2 Accuracy = 0.9999294274553672
```



۴. رگرسیون درجه پنج

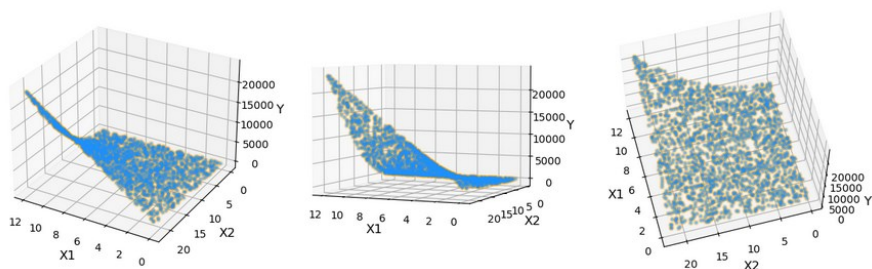
در این قسمت به بررسی نتایج روی رگرسیون درجه ۵ میپردازیم. انتظار ما به صورت تئوری این است که چون درجه ی رگرسیون از درجه ی داده های ما بیشتر است مقدار خطا در داده های آموزش ثابت بماند و یا حتی از حالت درجه ۳ کمتر شود، اما مقدار کمی **overfit** داریم که البته چون داده های زیادی داریم این مقدار زیاد نیست. بنابراین انتظار داریم مقدار کمی داده های تست خطا بیشتر شود ولی در داده های آموزشی خطا در مقایسه با رگرسیون درجه سه ثابت بماند و یا بهتر شود. در ادامه مشاهدات خود را خواهیم آورد.

• رگرسیون درجه پنج معادله نرمال (بسته)

در این قسمت ما با توجه به نتایج و استدلال های گذشته انتظار داریم در این حالت دقت بالاتری از رگرسیون با گرادیان نزولی بگیریم. انتظار ما این است که دلیل بالاتر بودن درجه چند جمله ای رگرسیون از چند جمله ای توزیع داده ها، انتظار **overfit** کمی داریم. اما نتایج به صورت زیر است:

```
Error Value on Test Data : 4.6080138743303176e-09 , R^2 Accuracy = 1.0
Error Value on Training Data : 1.4143615044454649e-08 , R^2 Accuracy = 1.0
```

مشاهده می شود دقت روی داده های آموزشی در این روش بالاتر از دقت روی داده های آموزشی معادله نرمال درجه سه است.



این دلایل زیادی می تواند داشته باشد. از جمله غلط بودن کد معادله نرمال ، مشکلات مربوط به محاسبه ی معکوس در معادله نرمال .

برای حل این مشکل داده ها را نرمال کردیم . خطا روی داده ی آموزش به حالت $1.232223516106237e-12$ رسید. ظاهرا مشکل حل شده است.

برای تست نتیجه نرمال سازی برای همین داده ها چند جمله ای درجه ی ۲۰ فیت می کنیم ، انتظار داریم دقت روی داده های آموزشی کمتر شود اما در کمال ناپاوری 27850.102972699773 خطای ما خواهد بود که خطای بیشتر نسبت به خطای چند جمله ای درجه ۵ است. با توجه به پرسش و پاسخ و جستجو هایی که انجام شد مشخص شد دلیل این است که چون در تولید فیچر ها تمام روابط بین متغیر هارا در نظر میگیریم پس عملا برای یک چند جمله ای درجه ۲۰ ، ۲۳۱ فیچر تولید میکنیم که مقدار بالایی است. میزان قابل توجهی از اینها روابطی هستند که وابسته خطی هستند و در محاسبه ی عددی وارون موجود در معادله نرمال، بد عمل میکنند و دقت را کاهش میدهند و عملا نتیجه مورد انتظار ما در تئوری را تولید نمیکند که راه حلی برای این مسئله در زمان محدودی که داشتم پیدا نکردم . پس یعنی مشکل ما به خاطر این است که معکوس را تحلیلی حساب نمیکنیم و عددی و تقریبی حساب میکنیم. ممکن است ماتریس مورد نظر اصلا معکوس نداشته باشدو دچار مشکل شویم. البته با کتابخانه آماده sklearn این مشکل وجود نداشت و نتایج قابل استدلال بود.

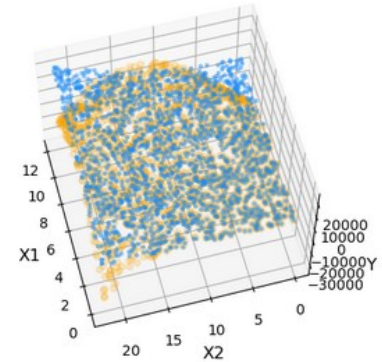
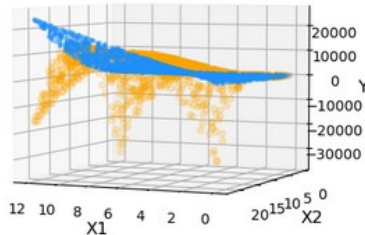
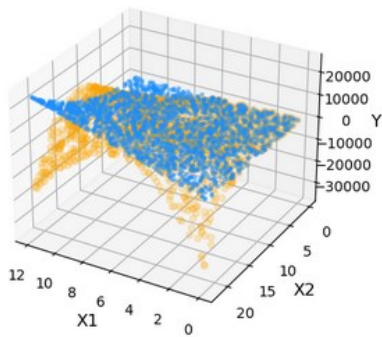
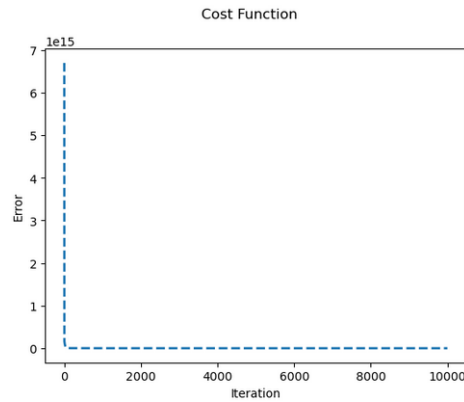
• رگرسیون درجه پنج با گرادبان نزولی

با توجه به توضیحات در مراحل قبلی انتخاب پارامتر های گرادبان نزولی کار پیچیده و زمانبر است. چون درجه ی ۵ است بنابراین باز هم باید نرخ یادگیری را کمتر کنیم ، چون سرعت همگرایی بیشتر میشود و ما نباید گرفتار واگرایی شویم. با توجه به آزمون و خطا مقدار 10^{-6} را انتخاب کردیم که بسیار کوچک است. بنابراین سرعت همگرایی ما به شدت کم خواهد بود. با تعداد تکرار ۱۰۰۰۰ ، دقت های ما به صورت زیر خواهد بود که نشان می دهد این تعداد تکرار کافی نیست و باید تعداد تکرار خیلی بیشتری داشته باشیم برای اینکه بتوانیم با این نرخ یادگیری دقت خوبی داشته باشیم.

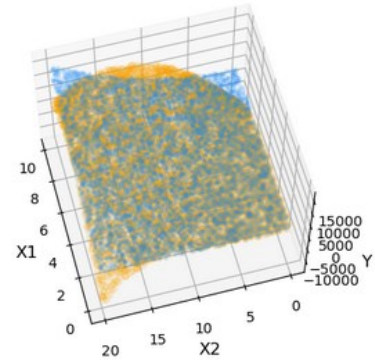
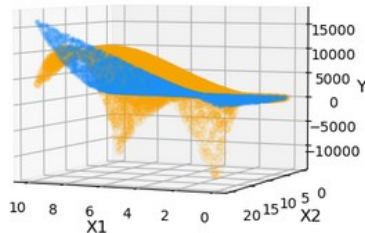
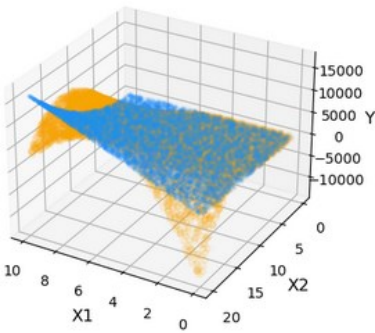
Error Value on Test Data : 87733544217.29219 , R^2 Accuracy = -0.7396980724361457
Error Value on Training Data : 50268191536.99261 , R^2 Accuracy = 0.4363277466636898

البته این تعداد تکرار به شدت کم است و فقط با شانس آوردن که انتخاب اولیه W مناسب باشد، چنین دقت هایی

میگیریم.



روی داده های تست



روی داده های آموزشی

که البته راه حل هایی برای حل این مشکل میتوان به بالا بردن تکرار الگوریتم (افزایش بسیار زیاد هزینه) ، پویا کردن مقدار نرخ یادگیری که از اور فیت جلوگیری شود (ابتدا بزرگ باشد و سپس کم بشود) و نرمال سازی داده هاست.

جدول مقدار خطا داده تست SSE	رگرسیون خطی	رگرسیون درجه ۳	رگرسیون درجه ۵
Normal Equation	12944759955.77201	5.309413410151504e-15	4.6080138743303176e-09
Gradient Descent	15894720003.55463	3966555.3221134352	87733544217.29219

۵. رگرسیون با استفاده از K Fold Cross Validation و Regularization

در این قسمت با استفاده از فرم بسته رگرسیون داده ها را Train کردیم. البته از 5 Fold Cross Validation استفاده کردیم به این معنی که داده های آموزشی که شامل ۸۰۰۰ خط است، به ۵ قسمت ۱۶۰۰ تایی تقسیم کردیم. البته لازم به ذکر است ابتدا داده ها را shuffle کرده و سپس به ۵ قسمت تقسیم کردیم. هر بار یک قسمت را به عنوان داده validation و بقیه را به عنوان آموزش.

1600 (Validation)	1600 (Training)	1600 (Training)	1600 (Training)	1600 (Training)
1600 (Training)	1600 (Validation)	1600 (Training)	1600 (Training)	1600 (Training)
1600 (Training)	1600 (Training)	1600 (Validation)	1600 (Training)	1600 (Training)
1600 (Training)	1600 (Training)	1600 (Training)	1600 (Validation)	1600 (Training)
1600 (Training)	1600 (Training)	1600 (Training)	1600 (Training)	1600 (Validation)

سپس لامبدا های ذکر شده را در فرمول رگرسیون با معادله نرمال دخیل میکنیم به شکل $\bar{w} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T y$ و سپس میانگین خطا ها در validation ها را میانگین میگیریم. همینطور در داده Train و داده تست. سپس مقدار میانگین خطا ها را در validation به ازای لامبدا های مختلف بررسی میکنیم تا ببینیم چه لامبدایی خطای کمتری تولید میکند. نتایج به شکل زیر است:

• رگرسیون درجه ۱

البته که نمیتوان ادعا کرد که همیشه چنین مقداری است اما با توجه به خطایی که در محاسبه ی وارون ماتریس وجود دارد و اینکه داده ها به صورت رندوم shuffle میشوند ممکن است انتخاب بهترین پارامتر برای لامبدا در این حالت کمی دچار مشکل شود اما نتیجه کلی که میتوان از این تست گرفت این است که باید مقدار lambda مقدار کوچکی باشد تا مقدار ارور کمتر باشد. نتیجه یک نمونه که بهترین لامبدا 0.0001 است را مشاهده میکنید.

```

Loss for 0.0001 - Validation Set : 3664712996.9535685
Loss for 0.0001 - Test Set : 12944842509.298565
Loss for 0.0001 - Train Set : 14653690908.159801

Loss for 0.001 - Validation Set : 3667337433.331006
Loss for 0.001 - Test Set : 12945553186.150864
Loss for 0.001 - Train Set : 14652524730.148945

Loss for 0.01 - Validation Set : 3669159356.5488153
Loss for 0.01 - Test Set : 12946030830.56185
Loss for 0.01 - Train Set : 14651713315.804651

Loss for 0.1 - Validation Set : 3664900042.187786
Loss for 0.1 - Test Set : 12945741175.141142
Loss for 0.1 - Train Set : 14653609496.513819

Loss for 1 - Validation Set : 3670045186.0833144
Loss for 1 - Test Set : 12954222417.487278
Loss for 1 - Train Set : 14651342166.400768

Loss for 10.0 - Validation Set : 3674531606.7526603
Loss for 10.0 - Test Set : 13025910694.85733
Loss for 10.0 - Train Set : 14651564159.730839

Loss for 100.0 - Validation Set : 3708919396.0931807
Loss for 100.0 - Test Set : 13712093110.3612
Loss for 100.0 - Train Set : 14818971255.674442

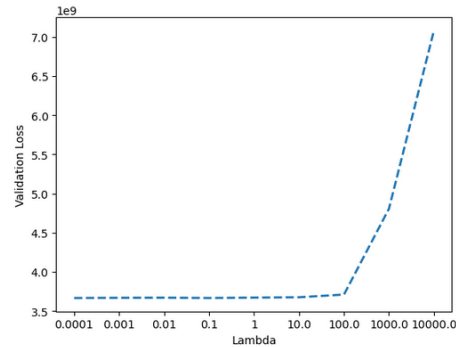
Loss for 1000.0 - Validation Set : 4805121314.934978
Loss for 1000.0 - Test Set : 18334129819.742352
Loss for 1000.0 - Train Set : 19197568796.0084

Loss for 10000.0 - Validation Set : 7080393956.60936
Loss for 10000.0 - Test Set : 24645025599.25639
Loss for 10000.0 - Train Set : 28292078587.476555

```

Best Lambda: 0.0001

شکل نمودار برای رگرسیون درجه ۱ و اعمال منظم سازی همیشه نموداری به شکل زیر است:



که یعنی هر چه مقدار λ بیشتر شود، مقدار ارور بیشتر خواهد شد یعنی مقدار بایاس زیاد شده و مقدار واریانس به طرز قابل توجهی پایین نیامده است. در واقع مدل ما زمانی که مقدار regularization زیاد میشود توان مدل سازی داده های ما را نخواهد داشت بنابراین دقت از چیزی که هست پایینتر خواهد آمد .

• رگرسیون درجه ۳

در این قسمت میدانیم که بهترین رگرسیون ممکن را داریم . بنابراین انتظار داریم مقدار لامبدای کمتر ، نتیجه بهتری برای ما تولید کند یعنی در $\lambda = 0.0001$ کمترین مقدار خطا را داشته باشیم و در $\lambda = 10000$ بیشترین مقدار خطا. تست های لازم انجام شد و نتیجه مورد قبول بود. میتوانید در ادامه نتایج را مشاهده بفرمایید:

```
Loss for 0.0001 - Validation Set : 3.3355946624229666e-11
Loss for 0.0001 - Test Set : 2.3503083044506126e-10
Loss for 0.0001 - Train Set : 1.3237659203948535e-10

Loss for 0.001 - Validation Set : 3.270135398443331e-09
Loss for 0.001 - Test Set : 2.2492195765213363e-08
Loss for 0.001 - Train Set : 1.3028423226684137e-08

Loss for 0.01 - Validation Set : 3.267194772327817e-07
Loss for 0.01 - Test Set : 2.2539188831204156e-06
Loss for 0.01 - Train Set : 1.3032107672571468e-06

Loss for 0.1 - Validation Set : 3.26264365823256e-05
Loss for 0.1 - Test Set : 0.00022503169262064237
Loss for 0.1 - Train Set : 0.00013024566210417606

Loss for 1 - Validation Set : 0.003259234942713658
Loss for 1 - Test Set : 0.022427131738769153
Loss for 1 - Train Set : 0.01298378634185933

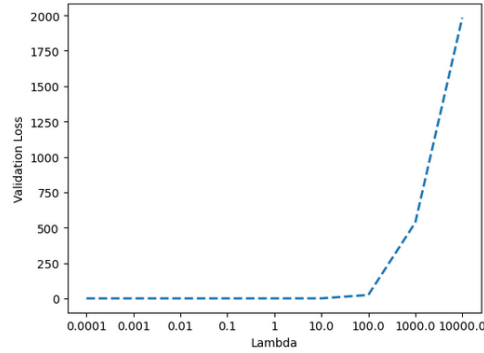
Loss for 10.0 - Validation Set : 0.3153480585715725
Loss for 10.0 - Test Set : 2.139338724556422
Loss for 10.0 - Train Set : 1.257292926759586

Loss for 100.0 - Validation Set : 24.229412319768805
Loss for 100.0 - Test Set : 154.9601741157698
Loss for 100.0 - Train Set : 96.70882366867832

Loss for 1000.0 - Validation Set : 535.5966177316079
Loss for 1000.0 - Test Set : 3363.762270647004
Loss for 1000.0 - Train Set : 2137.276675497841

Loss for 10000.0 - Validation Set : 1983.4869559070387
Loss for 10000.0 - Test Set : 10702.019685632209
Loss for 10000.0 - Train Set : 7921.022918685701
```

Best Lambda: 0.0001



• رگرسیون درجه ۵

در این قسمت نیز چون تعداد داده های ما زیاد است بنابراین نباید انتظار این را داشته باشیم واریانس مقدارش بسیار زیاد شود. بنابراین نمیتوانیم انتظار داشته باشیم مقدار بهینه برای λ مقدار زیادی تغییر کند و زیاد شود و احتمالا مشابه با دو حالت قبل خواهد بود و شاید کمی مقدار λ اضافه شود. البته باید توجه داشته باشیم در عمل مشکلات برای محاسبه عددی معادله نرمال داریم به خصوص قسمت وارون ماتریس وقتی روابط بین متغیر ها زیاد میشود (درجه رگرسیون و تعداد فیچر ها) مشکلات مربوط به محاسبات پیدا میکنیم و ممکن است اندکی با خطا روبرو شویم و انتظارات ما برآورده نشود. تست های لازم انجام شد و نتیجه ها به صورت زیر بود:

```

Loss for 0.0001 - Validation Set : 1.242808219331228e-06
Loss for 0.0001 - Test Set : 9.433212176708398e-05
Loss for 0.0001 - Train Set : 5.593216856491921e-06

Loss for 0.001 - Validation Set : 8.75141992146256e-06
Loss for 0.001 - Test Set : 0.0006179740946786651
Loss for 0.001 - Train Set : 2.9626523895940733e-05

Loss for 0.01 - Validation Set : 2.018890694888875e-05
Loss for 0.01 - Test Set : 0.0019423524371133514
Loss for 0.01 - Train Set : 8.304701361479681e-05

Loss for 0.1 - Validation Set : 0.0008897509758241957
Loss for 0.1 - Test Set : 0.08481845881436342
Loss for 0.1 - Train Set : 0.0035671963103295415

Loss for 1 - Validation Set : 0.08099241571044247
Loss for 1 - Test Set : 7.50021608987755
Loss for 1 - Train Set : 0.3191837477145267

Loss for 10.0 - Validation Set : 3.34201179402962
Loss for 10.0 - Test Set : 261.0436204006217
Loss for 10.0 - Train Set : 13.219101211736453

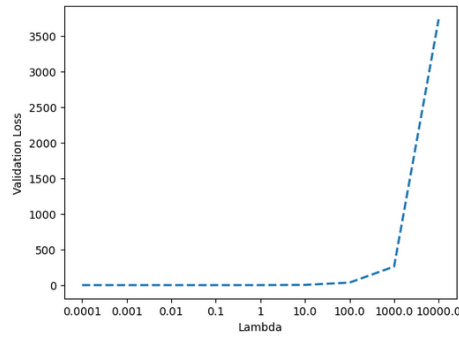
Loss for 100.0 - Validation Set : 34.822413948449466
Loss for 100.0 - Test Set : 1231.0478958511972
Loss for 100.0 - Train Set : 138.2150134585334

Loss for 1000.0 - Validation Set : 259.8863424854093
Loss for 1000.0 - Test Set : 3176.7590029883986
Loss for 1000.0 - Train Set : 1033.3777038311098

Loss for 10000.0 - Validation Set : 3729.828748928917
Loss for 10000.0 - Test Set : 67941.21842406617
Loss for 10000.0 - Train Set : 14839.82402540884

```

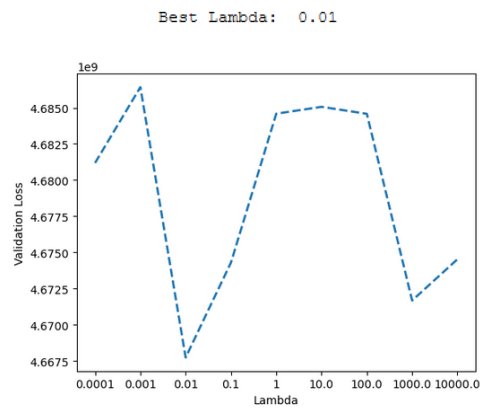
Best Lambda: 0.0001



• رگرسیون درجه ۲۰ (اضافه بر تمرین)

برای واضح شدن مشکلی که برای محاسبه ی وارون ماتریس داریم رگرسیون درجه ۲۰ را اضافه کردیم.

نتایج به صورت زیر بود:



به نظر میرسد کمی ناپایداری داریم . کد ها با دقت بررسی شدند و ایراد از کد ها نیست. احتمال بسیار قوی این مشکل از اینجا ناشی میشود که در $n=20$ ، وابستگی ها بین فیچر های تولید شده توسط کرنل خیلی زیاد است باعث میشود وارون ماتریس به درستی حساب نشود و همچنین به دلیل شافل کردن (دلیل اصلی به نظر نمیرسد) این مشکل رخ داده است.

البته باید دقت کنیم که بار ها اجرا شد اما در اکثر اجرا ها مقدار بهینه برای لامبدا 0.01 بود مثلا نمونه زیر:

