به نام خدا

دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر

پاسخ تمرین سری اول یادگیری ماشین

استاد:

دکتر احسان ناظرفرد

دانشجو:

حلیمه رحیمی

شماره دانشجویی:

۹۹۱۳۱۰۴۳

پاییز ۱۳۹۹

سوالات تشریحی

۱- مفاهیم زیر را تعریف کنید.

الف) یادگیری نظارتی (Supervised Learning)

در یادگیری نظارتی، ناظری وجود دارد که خروجی صحیح را در اختیار الگوریتم یادگیرنده قرار می دهد و الگوریتم به گونه ای باید نگاشتی هر چه صحیح تر از ورودی به خروجی بیابد. برای مثال در دسته بندی (Classification) با یادگیری نظارتی برچسب داده هایی را که برای آموزش الگوریتم یادگیرنده استفاده می شود داریم و عملکرد الگوریتم براساس آن بهبود می یابد.

ب) یادگیری نیمه نظارتی (Semi-Supervised Learning)

در این نوع یادگیری، تنها خروجی صحیح برخی از داده ها را داریم و باقی داده ها بدون برچسب اند.

پ) یادگیری بدون نظارت (Unsupervised Learning)

در این نوع یادگیری، ناظری وجود ندارد که برچسب های صحیح را در اختیار یادگیرنده قرار دهد و الگوریتم با استفاده از شباهت هایی که داده ها به یکدیگر دارند، به یادگیری می پردازد. به عبارتی فضای داده های ورودی ما ساختاری دارد که ممکن است الگوهای متفاوتی در آن یافت شود که با توجه به آنها بتوان اطلاعاتی از ورودی ها دریافت (به طور مثال اینکه گروهی از داده ها در یک دسته قرار می گیرند).

ت) یادگیری تقویتی (Reinforcement Learning)

اینگونه از یادگیری با پاداش و جریمه همراه است. الگوریتم بر اساس معیار کارایی خود پاداش یا جریمه دریافت می کند و به این صورت می آموزد که چه مجموعه اعمالی آن را به پاداش بیشتر می رساند.

ث) یادگیری عمیق (Deep Learning)

واژه ی «عمیق» در اینجا اشاره به عمق لایه های الگوریتم یادگیرنده دارد. یادگیرنده با استفاده از ویژگی های سطح بالایی که از طریق لایه ها به دست می آید به یادگیری می پردازد.

ج) رگرسیون (Regression)

نوعی یادگیری نظارتی است که در آن به دنبال پیشبینی یک مقدار عددی با توجه به داده های پیشین هستیم، بنابراین مقدار صحیح پاسخ برای داده های آموزش را داریم و الگوریتم به دنبال الگوهایی است که از از ویژگی های یک داده جدید، مقدار پاسخ مورد انتظار را پیشبینی کند.

چ) یادگیری برخط (Online Learning)

زمانی که تمام داده ها را نداریم و آن ها را در هنگام یادگیری پارامترها، یکی یکی دریافت می کنیم، یادگیری به صورت برخط انجام می گیرد.

ح) یادگیری فعال (Active Learning)

در این حالت یادگیرنده داده را برای برچسب دهی انتخاب می کند و ناظر برچسب صحیح را به الگوریتم می دهد.

خ) دسته بندی (Classification)

در دسته بندی، کلاس های داده های آموزش مشخص اند و الگوریتم با یادگیری الگوها از داده های آموزشی، به پیشبینی کلاس داده های تست می پردازد. در واقع در این حالت الگوریتم یاد می گیرد چه الگوهایی نشاندهنده ی این است که داده حاضر متعلق به کلاس بخصوصی باشد.

د) خوشه بندی (Clustering)

نوعی یادگیری بدون نظارت محسوب می شود که با توجه به شباهت هایی که بین داده ها هست، داده ها به گروه ها یا خوشه های مختلف تقسیم می شوند.

ذ) بیش برازش و کم برازش (Overfitting & Underfitting)

بیش برازش: گاهی پارامتر های تابع جداکننده به گونه ای انتخاب می شود که مدل به شدت روی داده های آموزش برازش شده است؛ به این معنی که به نظر می رسد نتیجه ی آن ها را حفظ کرده است و برای این داده ها خطای کمی را نشان می دهد. در حالیکه در برابر داده های جدید خطای بالایی دارد. در این حالت وجود یک داده ی پرت و یا نویز، تابع را به تناسب همان تغییر می دهد. معمولا وقتی تعداد پارامتر ها زیاد و تعداد داده ها کم باشد، این اتفاق می افتد. در این وضعیت، واریانس بالا است.

کم برازش: مدل به اندازه کافی روی داده ها برازش نشده است. در این حالت خطا هم برای داده های آموزش و هم برای داده های تست بالاست. وقتی مدل پیچیده نباشد و تعداد پارامترهای کمی داشته باشد، این حالت اتفاق می افتد. در این وضعیت، بایاس بالاست.

۲- همبستگی بین ویژگی ها به چه معنی است و چگونه می توان آن را تشخیص داد؟ (کامل توضیح دهید)

همبستگی از طریق فرمول ۱ به دست می آید و عددی بین ۱- و ۱ می باشد. هر چه پاسخ به این دو عدد نزدیکتر باشد، میزان همبستگی بیشتر است. اگر همبستگی برابر صفر باشد، به این معنی است که دو متغیر با هم همبستگی ندارند. به عبارتی با زیادتر شدن (یا کمتر شدن) یک متغیر، متغیر دیگر با نسبتی خطی به آن تغییر نمی کند. لازم است همبستگی بین ویژگی ها کم و همبستگی بین هر ویژگی و برچسب زیاد باشد. در صورتی که همبستگی بین ویژگی ها بالا باشد، نشان از این است که ویژگی اضافی است و تاثیر جدیدی بر پیشبینی نمی گذارد. علاوه بر محاسبه ی همبستگی بین دو ویژگی، می توان از طریق نمودار پراکندگی داده ها نسبت به دو ویژگی نیز متوجه همبستگی شد. در صورتی که داده ها به شکل یک بیضی مورب پراکنده اند که در هر نقطه با بالا رفتن یک متغیر، اندازه متغیر دیگر نیز بالا رفته، بین دو متغیر همبستگی وجود دارد.

$$Corr(x,y) = \frac{\sigma_{x,y}}{\sigma_x \sigma_y} \quad (1)$$

۳- معیارهای ارزیابی MSE، MAE، RMSE را با هم مقایسه کرده و بگویید در صورت داشتن داده های پرت و نویزی کدام یک بهتر عمل می کند؟ چرا؟

در MAE به خطاهای بزرگتر بهای بیشتری نسبت به خطاهای کوچکتر داده نمی شود و با تمامی خطاها به یک شکل برخورد می شود. درحالیکه در MSE به دلیل آنکه خطا به توان دو می رسد، خطاهای بزرگتر اثر بیشتری دارند. چنین چیزی باعث می شود در صورت وجود داده ی پرت با خطای بالایی مواجه شویم که در صورت نبود آن داده، بسیار کمتر می بود. چنین خطایی باعث می شود یادگیرنده به آن سمت تمایل پیدا کند. RMSE در مواجهه با داده های پرت و نویزی بهتر عمل می کند؛ چرا که با رادیکال گرفتن از MSE اثر خطای آنها بر کل خطا را کاهش می دهد.

۴- روش های گرادیان نزولی و معادله نرمال را با یکدیگر مقایسه کرده و برتری هر کدام را شرح دهید.

گرادیان نزولی نیاز به تعیین درجه یادگیری دارد و از طریق تکرار به پاسخ بهینه می رسد. این الگوریتم ممکن است نیاز به feature scaling داشته باشد. حال آنکه معادله نرمال نیازی به تعیین درجه یادگیری، تکرار و feature scaling ندارد. در معادله نرمال گاه ممکن است نیاز باشد با مسأله ی وارون ناپذیری مقابله کنیم. همچنین این روش در مقابل تعداد ویژگی های زیاد بسیار کندتر از گرادیان نزولی عمل می کند. در حالیکه گرادیان نزولی به علت اینکه به تعداد تکرار نیز بستگی دارد، در وضعیت ویژگی های کم بسیار کندتر از معادله نرمال عمل خواهد کرد؛ درست مثل حالتی که در بخش سوالات پیاده سازی این تمرین با آن مواجه می شویم.

۵- رگرسیون Lasso را توضیح داده و تفاوت آن را با رگرسیون خطی شرح دهید.

تابع به کار رفته در رگرسیون Lasso را در فرمول ۲ می بینید. این نوع رگرسیون با قرار دادن شرط از بالا رفتن وزن ها جلوگیری می کند و عمل رگولاریزیشن انجام می دهد. ضریبی که در پشت مجموع وزن ها قرار گرفته، میزان رگولاریزیشن را تغییر می دهد و البته می تواند موجب Feature Selecion شود؛ از آن جهت که به جای مجموع توان دوم وزن ها، از مجموع قدر مطلق آنها استفاده شده است بنابراین بالا رفتن ضریب موجب صفر شدن وزن ویژگی های کم اثر می شود.

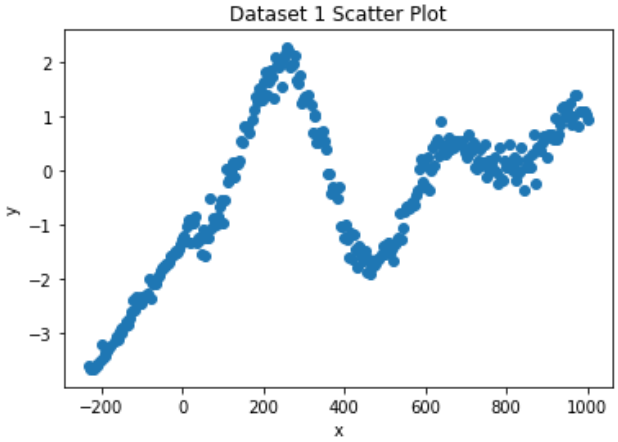
$$\sum_{\mathrm{i}=1}^{\mathrm{m}} (y_i - \sum_j x_{ij}\theta_j)^2 + \lambda \sum_{j=1}^{n} |\theta_j| \qquad \sum_{j=1}^{n} |\theta_j| \le t \quad (2)$$

سوالات پیاده سازی

بخش اول

مجموعه داده ی Dataset1.csv را بارگذاری کنید.

۱- داده ها را رسم کنید.



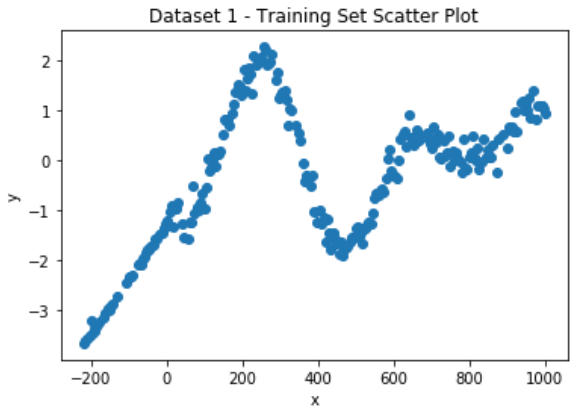
۲- شافل کردن و نرمال سازی داده ها به چه منظور انجام می شوند؟ آیا مجموعه داده ی Dataset1.csv نیازی به این اقدامات دارد؟ توضیح داده و در صورت لزوم این موارد را بر روی مجموعه داده اعمال کنید.

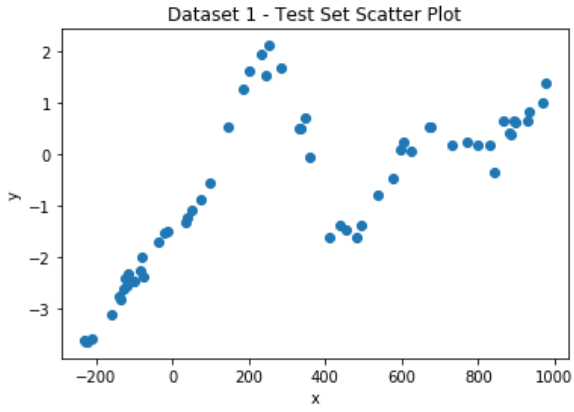
شافل کردن در مواقعی نیاز می شود که نیاز است داده ها به دلیلی تقسیم بندی شوند و یا ترتیب ورود آنها بسیار بر روند یادگیری تاثیرگذار است؛ به طور مثال وقتی قرار است داده ها را به مجموعه های مختلف آموزش، اعتبارسنجی و تست تقسیم کنیم، در بوت استرپینگ (Boot Strapping) و در استفاده از Stochastic Gradient Descent برای شبکه های عصبی. به بیان بهتر وقتی نیاز به انتخاب داده ها هست. این کار را به این دلیل انجام می دهیم که انواع داده ها با هم بُر بخورند و دیتاست حاصل برای هر مجموعه به سمت نوع خاصی از داده تمایل نداشته باشد.

با توجه به آنکه در سوال ها از ما MSE فاز آموزش و آزمون خواسته شده، دیتاست اول را به دو بخش مجموعه آموزش و مجموعه آزمون تقسیم کرده ام. بنابراین در اینجا برای جلوگیری از بایاس شدن دیتاست آموزشی یا آزمون از شافل کردن استفاده کرده ام.

نرمال سازی در صورتی انجام می گیرد که واحد مقادیر یک ویژگی با دیگری متفاوت باشد؛ برای مثال مقادیر یک ویژگی اعدادی تک رقمی و دیگری اعدادی سه رقمی باشد. در چنین حالت هایی ویژگی با واحد بزرگتر توزیع داده ها را تحت تاثیر قرار می دهد. در اینجا هم به علت استفاده از رگرسیون چندجمله ای و هم بخاطر رگولاریزیشن، واحد های ویژگی ها چند برابر یکدیگر هستند.

۳- تابعی بنویسید که با استفاده از روش گرادیان نزولی و با دریافت داده ها، درجه، تعداد تکرار، نرخ یادگیری و ضریب رگولاریزیشن یک خط/ منحنی بر روی داده ها برازش کند. با استفاده از این تابع به ازای پنج درجه ی مختلف یک نمودار بر روی نقاط برازش کنید (ضریب رگولاریزیشن را صفر قرار دهید). برای هر درجه، سه مقدار تکرار با فاصله ی مناسب انتخاب کرده و با توجه به آن مقادیر MSE را بیابید و گزارش کنید (به ازای هر درجه و تعداد تکرار مقدار حدودی مناسب را برای نرخ یادگیری بیابید و بر اساس آن برازش را انجام دهید).





در ابتدا مجموعه داده ها را به دو بخش آموزش و آزمون با نسبت ۰/۸ و ۰/۲ تقسیم کردم. در تصویر بالا پراکندگی هر یک از این دو مجموعه را می بینید.

در جدول زیر مقداردهی پارامتر های ورودی را آورده ام.

α		Degree				
		1	**2**	**4**	**8**	**12**
Number of Iteration	**5000**	0.03	0.05	0.5	0.99	0.99
	50000	0.03	0.05	0.5	0.99	0.99
	300000	0.03	0.05	0.5	0.99	0.99

همانطور که مشاهده می کنید مقدار α برای درجه ۸ و ۱۲ برابر و رقمی نسبتا بزرگ است؛ علت این انتخاب بنده، بالا بردن سرعت همگرایی بوده است؛ چرا که این درجات با حتی قرار دادن تعداد تکرار برابر سیصد هزار به بهینه ترین حالت خود نمی رسد. انتخاب مقداری کمتر موجب بیشتر شدن خطای آموزش و آزمون می شد. البته از حدود مقدار ۰/۲ برای درجه ۴، مقدار ۰/۴ برای درجه ۸ و مقدار ۰/۶ برای درجه ۱۲، سرعت کاهش خطا کمتر شد ولی به پیشرفت خود ادامه داد.

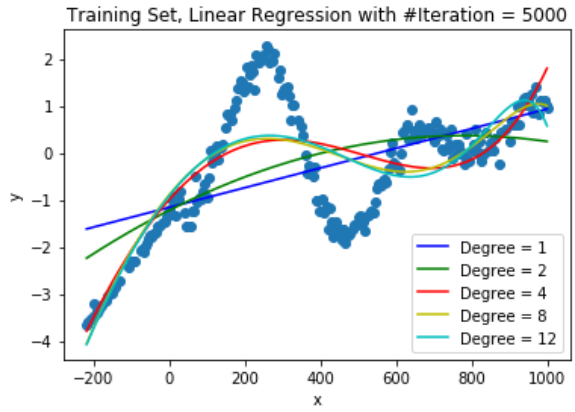
مقدار را با بالا رفتن تعداد تکرار کمتر نکرده ام؛ چرا که با مشاهده مقدار خطا در هر مرحله از یادگیری متوجه شدم الگوریتم دچار واگرایی نمی شود و با بالا رفتن مقدار درجه یادگیری تنها سرعت همگرایی بیشتر شده است.

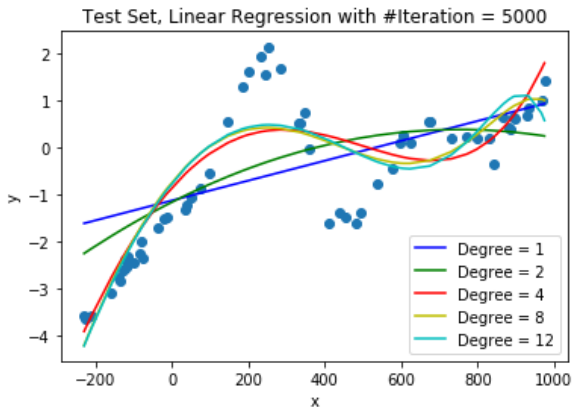
مقدار خطا را برای مجموعه آموزش و آزمون در جدول زیر مشاهده می کنید. در تعداد تکرار برابر ۵۰۰۰، خطا برای درجه ۱۲ بیشتر از خطای درجه ۸ می شود، علت آن است که با بالا رفتن درجه و پایین ماندن دفعات تکرار، تابع به درستی روی داده ها برازش نمی شود. در حالیکه در دفعات بالاتر، این مقدار کمتر می شود.

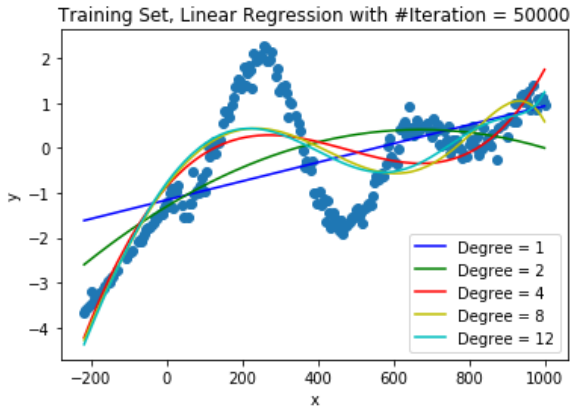
با توجه به خطای آموزش و آزمون برای درجه ۴ می توان متوجه شد که با بالا رفتن تکرار، خط به قدری روی داده های آموزش برازش شده که در مقابل داده های جدید ضعیف تر از قبل عمل می کند. با چندین بار اجرا گرفتن متوجه شدم که متناسب با داده های تست اینکه کدام یک از درجات دچار این حالت شوند تغییر می کند.

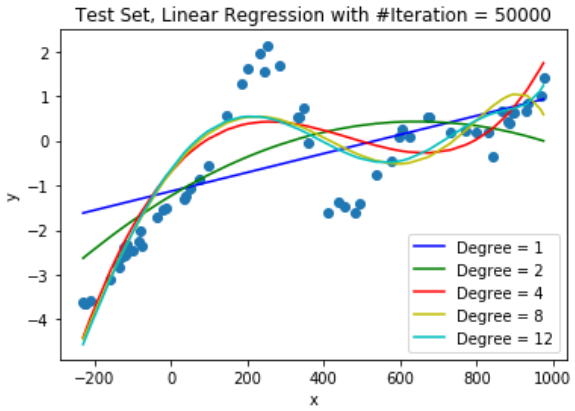
از آن جهت که توزیع داده های آموزش و آزمون شباهت زیادی دارد، در باقی حالت ها با بالا رفتن درجه و تعداد تکرار، خطای آزمون نیز به همراه خطای آموزش کاهش یافته است.

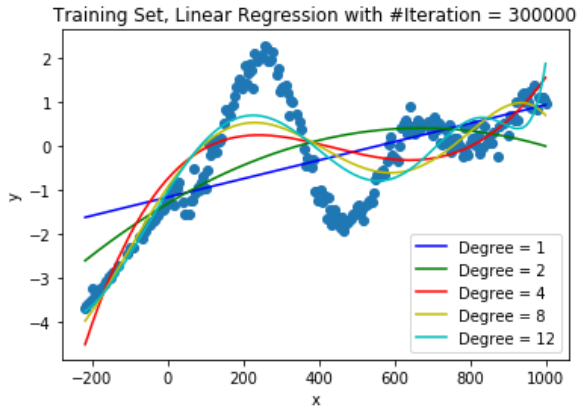
	D	Training Set MSE	Test Set MSE
Number of Iteration = 5000	1	1.3329810708099803	1.3694293912754523
	2	1.1762328229587824	1.060626618922919
	4	0.7458668666106968	0.572300422575138
	8	0.6668904915551802	0.5409508859272895
	12	0.6804802706166124	0.5651535579931687
Number of Iteration = 50000	1	1.3329810679919434	1.369388800788732
	2	1.1582134285572458	0.9685114438458106
	4	0.7210752034621795	0.5907697831212017
	8	0.6318154433969158	0.5423285887075443
	12	0.5907546937422282	0.5003411446325391
Number of Iteration = 300000	1	1.3329810679919434	1.369388800788732
	2	1.158213428557179	0.9685113183306272
	4	0.7134315801354657	0.6339719534982807
	8	0.5693248510175506	0.4528281466271646
	12	0.47973121253887496	0.35486832870766455

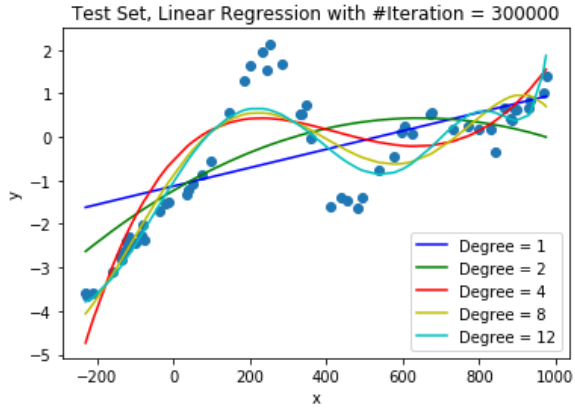






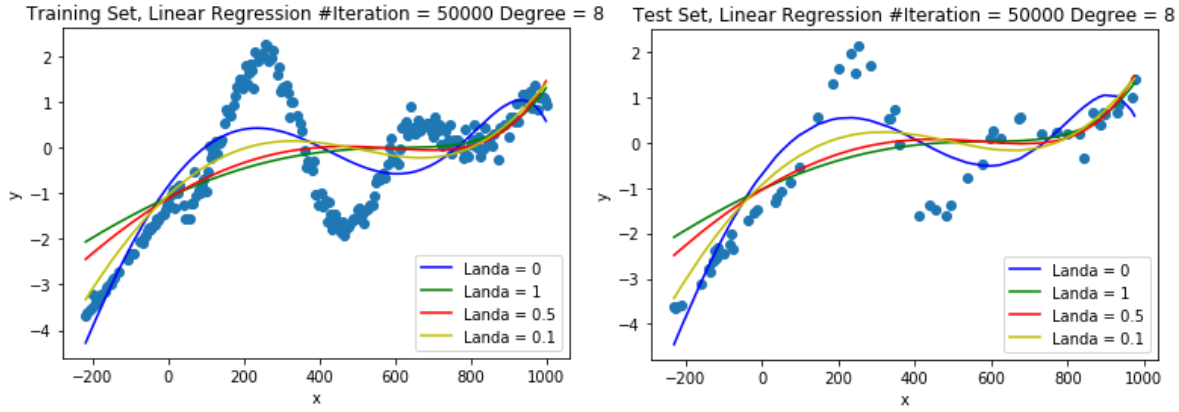






همانطور که در تصویر مشخص است منحنی درجات بالاتر با بیشتر شدن تکرار به داده های آموزش نزدیکتر شده اند، در حالیکه منحنی درجات پایین تر به علت بایاس بالا، تقریبا به شکل قبلی خود باقی مانده اند.

۴- به ازای بهترین مقادیر یافته شده برای پارامتر ها در مرحله ی قبل و به ازای ۳ مقدار مختلف با فاصله ی مناسب برای ضریب رگولاریزیشن نمودار را بر روی نقاط برازش کرده و به همراه داده ها رسم کنید. مقدار خطای MSE برای هر دو فاز آموزش و آزمون و بردار ضرایب θ را گزارش کنید. تغییر ضریب رگولاریزیشن چه تاثیری بر روی اندازه بردار ضرایب θ دارد؟



درجه ۸ را به این دلیل انتخاب کردم که با تکرار بیشتر، مقدار خطا برای آن کاهش می یافت. دلیل انتخاب ۵۰۰۰۰ تکرار و همچنین انتخاب نکردن درجه ۱۲ سرعت آموزش بود.

با بالاتر رفتن ضریب رگولاریزیشن قدر مطلق مقادیر بردار ضرایب θ کاهش پیدا می کند که باعث می شود منحنی درجات غیر یک به خط صاف نزدیکتر شوند و همچنین به علت نزدیکتر شدن این ضرایب به صفر، خط یا منحنی افقی تر می شود. در اینجا از آن جهت که به درجه ی ۸ پرداختم، می توانید در تصویر مشاهده کنید که با بالاتر رفتن ضریب رگولاریزیشن، منحنی پیچیدگی های خود را از دست داده است.

همانطور که در جدول زیر مشاهده می کنید قدر مطلق مقادیر θ با کاهش λ نسبتا افزایش پیدا کرده و با افزایش λ به صفر نزدیکتر شده است. البته در مقادیر نزدیکتر λ این اتفاق کمتر رخ داده ولی با تغییر بیشتر آن متوجه تفاوت چشمگیری می شویم.

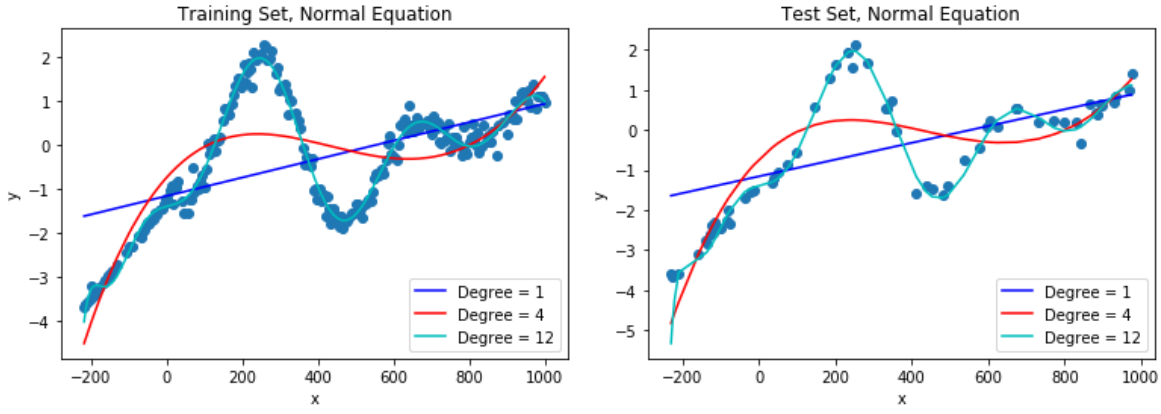
λ	1	0.5	0.1	0
θ_0	1.91343631	-2.1940669	-2.72157977	-3.06251504
θ_1	4.66766138	6.10846299	9.37264612	11.95542543
θ_2	-3.10599036	-5.25556494	-12.99178238	-21.35164638
θ_3	-0.91530935	-1.12147946	-0.03784025	2.08843222
θ_4	-0.03375977	0.40178391	4.23933234	9.55900101
θ_5	0.52476309	1.10118987	4.33929665	8.42986512

θ_6	0.70213797	1.09734931	2.24596751	3.33079451
θ_7	0.71930443	0.83511532	-0.32962087	-2.42369096
θ_8	0.66588558	0.49490832	-2.70717618	-7.52381841

در جدول زیر مقدار خطا را برای هر دو مجموعه آموزش و آزمون آورده ام. از آن جهت که مجموعه آزمون، توزیعی مشابه مجموعه آموزش دارد، با کمتر شدن λ خطای مربوطه کاهش پیدا کرده است. در صورتی که داده ها با واریانس بیشتری حول منحنی پراکنده بودند احتمالا با این میزان از خطا مواجه نمی شدیم. رگولاریزیشن در این شرایط خوب عمل نکرده است اما در شرایط دیگر ممکن است موجب بهبود یادگیری شود.

λ	Training Set MSE	Test Set MSE
1	1.2467135971166705	1.6779589410256972
0.5	1.13988645861043	1.4967892066079558
0.1	0.9085431027423044	1.1324923633732102
0	0.6318154433969158	0.5423285887075443

۵- تابعی بنویسید که با دریافت داده ها و درجه، یک خط/ منحنی به روش معادله نرمال بر روی داده ها برازش کند، سه درجه ی مختلف با فاصله مناسب را امتحان کنید و نمودار خط را همراه با داده ها رسم کرده و نتایج را بررسی کنید.



در اینجا درجه ها را به ترتیب، ۱، ۴ و ۱۲ انتخاب کردم. همانطور که مشاهده می کنید مقداردهی به ضرایب θ به گونه ای بوده که میزان خطا برای داده های آزمون از آنچه در روش گرادیان نزولی با تعداد تکرار ۵۰۰۰ به دست می آمد، بیشتر شده است. به این معنی است که در بهینه ترین حالت تابع هزینه برای داده های آموزش، خط خروجی را به شکلی تغییر می دهد که بر داده های آموزش کاملا برازش شده و میزان تعمیم آن کم تر است.

در درجه ی ۴ به بهینه ترین حالت منحنی برای داده های آموزش رسیده ایم و همانطور که می بینید خطا برای داده های آزمون نیز کاهش یافته است؛ از آن جهت که داده های آزمون توزیعی تقریبا مشابه داده های آموزش دارد.

در درجه ی ۱۲ می توان گفت منحنی کاملا بر داده های آموزش برازش شده است. کم تر شدن میزان خطای داده های آزمون نیز به دلیل داشتن توزیع مشابه می باشد.

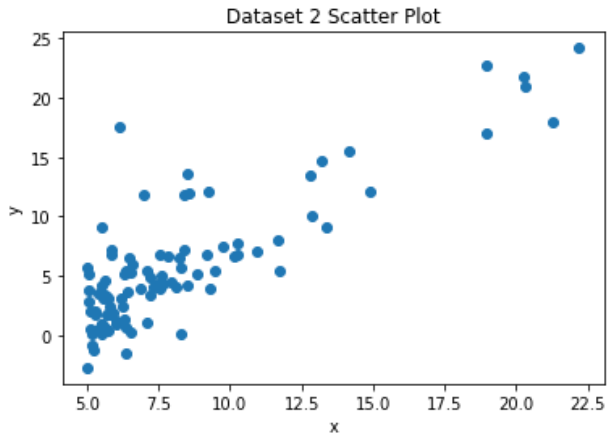
مقادیر مرتبط با پارامتر ها و مقدار خطا در جدول زیر آورده شده است.

Degree	Training Set MSE	Test Set MSE
1	1.3329810679919432	1.359012112747184
4	0.7134267209305873	0.6073132045702896
12	0.038612606335732075	0.11809515648650504

بخش دوم

مجموعه داده ی Dataset2.csv را بارگذاری کنید.

۱- داده ها را رسم کنید.

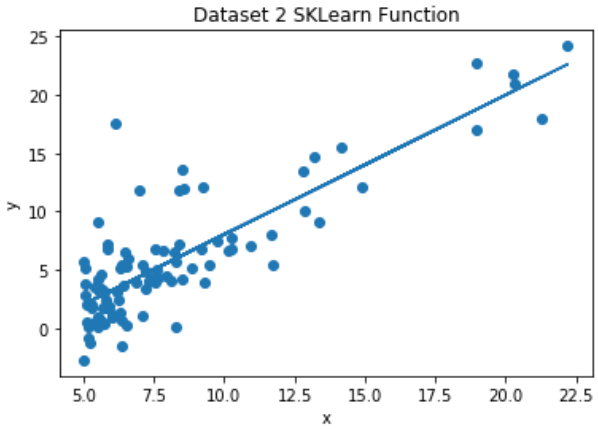


۲- با استفاده از تابعی که در قسمت ۳ بخش اول نوشته اید و با درجه ی ۱، خطی را بر روی نقاط برازش کنید.

MSE = 8.953942751950356

۳- با استفاده از یک کتابخانه ی آماده، با استفاده از LinearRegression خطی را بر روی نقاط برازش کنید.

برای این منظور از کتابخانه ی SKLearn استفاده کردم.



MSE = 8.953942751950358

مشاهده می شود که هر دو تابع برنامه، خطی مشابه را برازش می کنند و خطای آنها با هم برابر است.