HOMEWORK 5

Steven Kan
nedId: pkan2
id: 9075859844

Instructions: Although this is a programming homework, you only need to hand in a pdf answer file. There is no need to submit the latex source or any code. You can choose any programming language, as long as you implement the algorithm from scratch.
Use this latex file as a template to develop your homework. Submit your homework on time as a single pdf file to Canvas. Please check Piazza for updates about the homework.

Linear Regression (100 pts total, 10 each)

The Wisconsin State Climatology Office keeps a record on the number of days Lake Mendota was covered by ice at http://www.aos.wisc.edu/~sco/lakes/Mendota-ice.html. Same for Lake Monona: http://www.aos.wisc.edu/~sco/lakes/Monona-ice.html. As with any real problems, the data is not as clean or as organized as one would like for machine learning. Curate two clean data sets for each lake, respectively, starting from 1855-56 and ending in 2018-19. Let x be the year: for 1855-56, $x = 1855$; for 2017-18, $x = 2017$; and so on. Let y be the ice days in that year: for Mendota and 1855-56, $y = 118$; for 2017-18, $y = 94$; and so on. Some years have multiple freeze thaw cycles such as 2001-02, that one should be $x = 2001, y = 21$.

1. Plot year vs. ice days for the two lakes as two curves in the same plot. Produce another plot for year vs. $y_{Monona} - y_{Mendota}$.

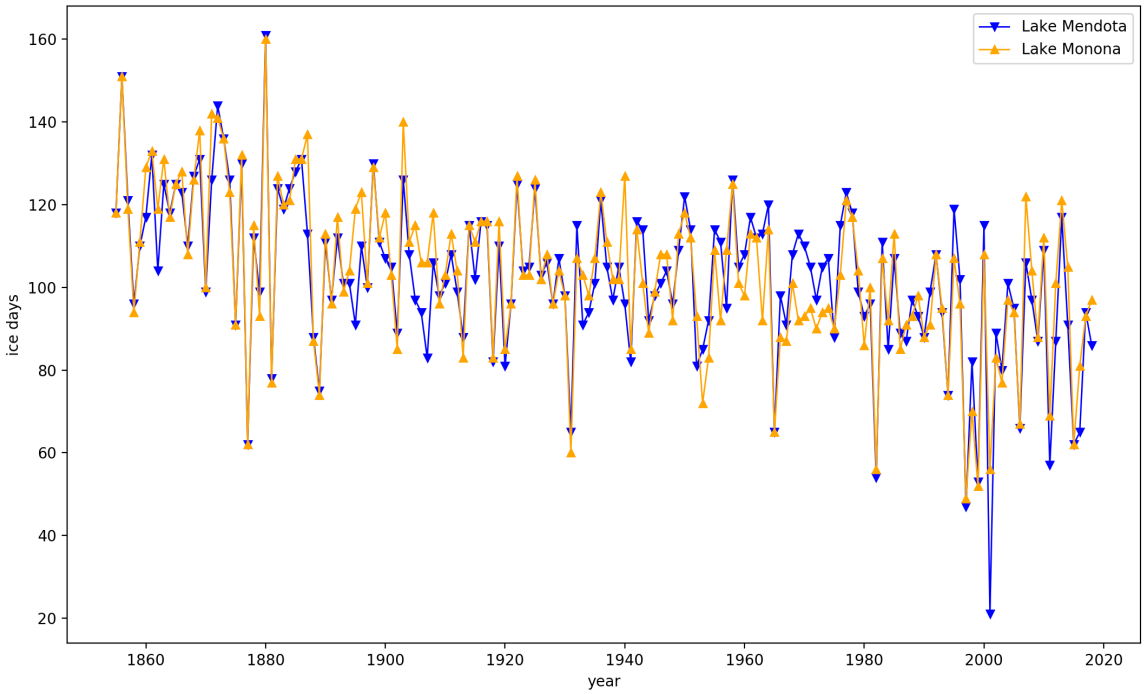

Figure 1: Year vs. ice days

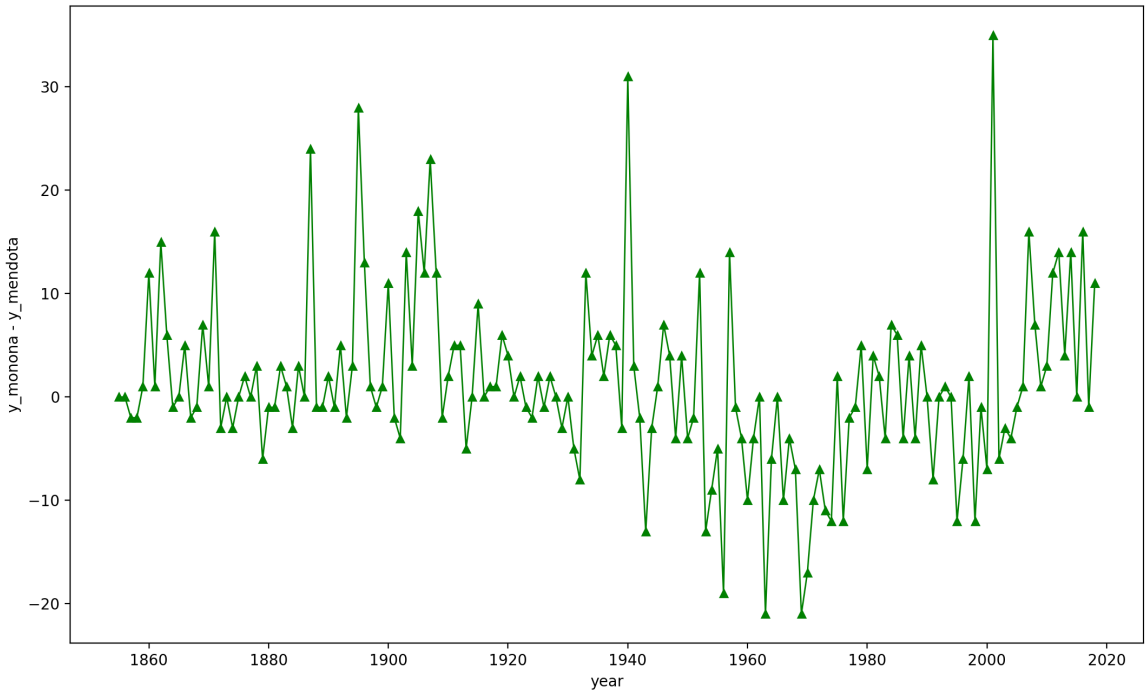

Figure 2: Year vs. $y_{Monona} - Y_{Mendota}$

2. Split the datasets: $x \leq 1970$ as training, and $x > 1970$ as test. (Comment: due to the temporal nature this is NOT an iid split. But we will work with it.) On the training set, compute the sample mean $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and the sample standard deviation $\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}$ for the two lakes, respectively.

Lake	Average	standard deviation
Mendota	107.1896551724138	16.74666159754441
Monona	108.48275862068965	18.122521543826252

Table 1: Average and Standard Deviation for the training sets

3. Using training sets, train a linear regression model

$$\hat{y}_{Mendota} = \beta_0 + \beta_1 x + \beta_2 y_{Monona}$$

to predict $y_{Mendota}$. Note: we are treating y_{Monona} as an observed feature. Do this by finding the closed-form MLE solution for $\beta = (\beta_0, \beta_1, \beta_2)^\top$ (no regularization):

$$\min_{\beta} \frac{1}{n}\sum_{i=1}^{n}(x_i^\top \beta - y_i)^2.$$

Give the MLE formula in matrix form (define your matrices), then give the MLE value of $\beta_0, \beta_1, \beta_2$.
Let's define **design matrix** X:

$$X_{n\times 3} = \begin{pmatrix} -x_1^\top - \\ -x_2^\top - \\ \vdots \\ -x_i^\top - \\ \vdots \\ -x_n^\top - \end{pmatrix}$$

where, vector $x_i = \begin{pmatrix} 1 \\ x \\ y_{Monona} \end{pmatrix}$, for the training point i.

$$Y_{n\times 1} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix}$$

where, y_i is the value of $y_{Mendota}$ for training point i.
Then, our target function becomes:

$$\min_{\beta} \frac{1}{n} \cdot ||X\beta - Y||_2^2$$

We take the gradient of the loss function:

$$\nabla_\beta \frac{1}{n} \cdot ||X\beta - Y||_2^2 = \frac{2}{n} \cdot X^\top (X\beta - Y) = 0$$

$$\hat{\beta} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \end{pmatrix} = (X^\top X)^{-1} X^\top Y \quad \text{, if } (X^\top X) \text{ is invertible.}$$

As we plug in the training set, we can get:

$\hat{\beta}_0$	-6.41827663e+01
$\hat{\beta}_1$	4.12245664e-02
$\hat{\beta}_2$	8.52950638e-01

Table 2: MLE value of $\beta_0, \beta_1, \beta_2$

□

4. Using the MLE above, give the (1) mean squared error and (2) R^2 values on the Mendota test set. (You will need to use the Monona test data as observed features.)

Mean Square Error	124.26409483939055
R^2	0.710490071562383

Table 3: MSE and R^2 value for Mendota Test Set

5. "Reset" to Q3, but this time use gradient descent to learn the β's. Recall our objective function is the mean squared error on the training set:

$$\frac{1}{n} \sum_{i=1}^{n} (x_i^\top \beta - y_i)^2.$$

Derive the gradient.
Solution:

$$\nabla_\beta \frac{1}{n} \sum_{i=1}^{n} (x_i^\top \beta - y_i)^2 = \frac{2}{n} \sum_{i=1}^{n} (x_i^\top \beta - y_i) x_i = \frac{2}{n} \sum_{i=1}^{n} (x_i x_i^\top \beta - y_i x_i)$$

$$= \begin{pmatrix} \frac{2}{n} \sum_{i=1}^{n} (\beta_0 \cdot x_{i0} + \beta_1 \cdot x_{i1} + \beta_2 \cdot x_{i3} - y_i) \cdot x_{i0} \\ \frac{2}{n} \sum_{i=1}^{n} (\beta_0 \cdot x_{i0} + \beta_1 \cdot x_{i1} + \beta_2 \cdot x_{i3} - y_i) \cdot x_{i1} \\ \frac{2}{n} \sum_{i=1}^{n} (\beta_0 \cdot x_{i0} + \beta_1 \cdot x_{i1} + \beta_2 \cdot x_{i3} - y_i) \cdot x_{i2} \end{pmatrix}$$

Or similar to Q3, to express in the matrix form:

$$\nabla_\beta \frac{1}{n} \cdot ||X\beta - Y||_2^2 = \frac{2}{n} \cdot X^\top (X\beta - Y)$$

□

6. Implement gradient descent. Initialize $\beta_0 = \beta_1 = \beta_2 = 0$. Use a fixed stepsize parameter $\eta = 0.1$ and print the first 10 iteration's objective function value. Tell us if further iterations make your gradient descent converge, and if yes when; compare the β's to the closed-form solution. Try other η values and tell us what happens. **Hint:** Update $\beta_0, \beta_1, \beta_2$ simultaneously in an iteration. Don't use a new β_0 to calculate β_1, and so on.
For $\eta = 0.1$:

iteration	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	objective value
0	0	0	0	11767.65517
1	21.43793	40964.99310	2379.04828	6180350808297759.0
2	-15720688.41776	-30074879522.34012	-1703451106.72243	3.33063e+27
3	11540587855812.344	2.20780e+16	1250503915942610.2	1.79489e+39
4	-8.47197e+18	-1.62075e+22	-9.17997e+20	9.67279e+50
5	6.21929e+24	1.18980e+28	6.73904e+26	5.21272e+62
6	-4.56559e+30	-8.73433e+33	-4.94714e+32	2.80917e+74
7	3.35161e+36	6.41189e+39	3.63171e+38	1.51388e+86
8	-2.46042e+42	-4.70698e+45	-2.66604e+44	8.15836e+97
9	1.80620e+48	3.45540e+51	1.95715e+50	4.39659e+109
10	-1.32594e+54	-2.53662e+57	-1.43675e+56	2.36935e+121

Table 4: For learning rate $\eta = 0.1$

For the case of $\eta = 0.1$, the objective value diverges!

For $\eta = 10^{-2}$:

iteration	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	objective value
0	0	0	0	11767.65517
1	2.143793103448276	4096.499310344827	237.9048275862069	61801992682736.33
2	-157203.02535005938	-300741421.5246425	-17034082.838534597	3.33046e+23
3	11540163402.434479	22077218083140.258	1250457923340.8376	1.79476e+33
4	-847155344821240.6	-1.6206731781732073e+18	-9.179524380515606e+16	9.671839e+42
5	6.2189082878412095e+19	1.1897248741020536e+23	6.738624810361575e+21	5.21208e+52
6	-4.5652571903138915e+24	-8.733687304263348e+27	-4.9467774638962895e+26	2.80875e+62
7	3.351323455671588e+29	6.411338922893509e+32	3.631394826980905e+31	1.51362e+72
8	-2.460183169606338e+34	-4.706519177088443e+37	-2.665781609475721e+36	8.15676e+81
9	1.8060033022987922e+39	3.4550228947035804e+42	1.9569316827295062e+41	4.39562e+91
10	-1.325774425339251e+44	-2.536308204380995 8e+47	-1.4365698965204053e+46	2.36877e+101

Table 5: For learning rate $\eta = 10^{-2}$

In the case of $\eta = 10^-2$, the objective loss value still diverges!

For $\eta = 10^{-3}$:

iteration	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	objective value
0	0	0	0	11767.65517
1	0.2143793103448276	409.64993103448273	23.79048275862069	617868397288.2198
2	-1571.644370741974	-3006676.845370565	-170298.0055163805	3.32883e+19
3	11535919.441691762	22069099060.20844	1249998060.9150717	1.79344e+27
4	-84673997533.31728	-161987854447060.1	-9175023559778.639	9.66236e+34
5	621509702342816.0	1.18899574997542e+18	6.734495037137094e+16	5.20570e+42
6	-4.561900008963962e+18	-8.727264758739253e+21	-4.943139721628609e+20	2.80462e+50
7	3.348448401261865e+22	6.405838723200372e+25	3.628279503184544e+24	1.51102e+58
8	-2.4577712518647437e+26	-4.701904993378621e+29	-2.663168126854369e+28	8.14078e+65
9	1.804011530897824e+30	3.4512124831818513e+33	1.954773458237698e+32	4.38593e+73
10	-1.3241499188108372e+34	-2.5332003987412587e+37	-1.4348096293657396e+36	2.36296e+81

Table 6: For learning rate $\eta = 10^{-3}$

In the case of $\eta = 10^-3$, the objective loss value still diverges!

For $\eta = 10^{-4}$:

iteration	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	objective value
0	0	0	0	11767.65517
1	0.02143793103448276	40.96499310344827	2.379048275862069	6163541503.948575
2	-15.677855431557674	-29993.03146611944	-1698.6977682672534	3312533871817738.0
3	11493.537137854139	21988018.330316674	1245405.795853439	1.78028834861592e+21
4	-8425946.762745045	-16119482693.130024	-913010632.5043329	9.56798e+26
5	6177087050.983098	11817241505903.053	669330855689.1824	5.14222e+32
6	-4528441195905.346	-8663255482012027.0	-490688474288211.7	2.76363e+38
7	3319813934224766.0	6.351058790591787e+18	3.5972520428849734e+17	1.48529e+44
8	-2.4337656339312415e+18	-4.6559804042839154e+21	-2.63715635033227e+20	7.98253e+49
9	1.7842009456737557e+21	3.4133133135515917e+24	1.933307294898436e+23	4.29013e+55
10	-1.308003108500241e+24	-2.5023103116475455e+27	-1.4173134239980034e+26	2.30569e+61

Table 7: For learning rate $\eta = 10^{-4}$

In the case of $\eta = 10^{-4}$, the objective loss value still diverges!

For $\eta = 10^{-5}$:

iteration	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	objective value
0	0	0	0	11767.655172413792
1	0.002143793103448276	4.096499310344828	0.2379048275862069	60131653.12688203
2	-0.15291972672936985	-292.55661590257375	-16.55874899301736	315283852895.1312
3	11.075444458443462	21188.160974055587	1200.1190648412576	1653112656695485.5
4	-801.9736350515137	-1534237.1192611272	-86899.46536642147	8.667686065008162e+18
5	58071.15623446466	111094577.74216975	6292418.504590769	4.544686136015049e+22
6	-4204949.980038337	-8044391964.108703	-455635924.3174436	2.382893418148727e+26
7	304481699.5881958	582496854660.441	32992735061.875473	1.2494110423289576e+30
8	-22047611939.084602	-42178773386580.836	-2389013922466.2954	6.550976811654033e+33
9	1596474247463.7212	3054177735318040.5	172989220537240.5	3.434842156255704e+37
10	-115601183015046.72	-2.2115393336402995e+17	-1.25262017691333e+16	1.8009742634720842e+41

Table 8: For learning rate $\eta = 10^{-5}$

In the case of $\eta = 10^{-5}$, the objective loss value still diverges!

For $\eta = 10^{-6}$:

iteration	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	objective value
0	0	0	0	11767.655172413792
1	0.00021437931034482758	0.4096499310344827	0.02379048275862069	461425.3207670699
2	-0.0011433145086730083	-2.188196283163668	-0.12276462096465635	18541573.033002064
3	0.007467553560888404	14.284581677099885	0.8108437151332372	745520179.8162065
4	-0.04713253210483675	-90.17004475348105	-5.104900056550123	29976359085.269817
5	0.2990899499338722	572.179656943501	32.41130193333086	1205309353426.9465
6	-1.896315591633006	-3627.800073159465	-205.47578055352147	48463879387659.984
7	12.024819019151845	23004.398155432176	1302.9776886517266	1948667866199789.5
8	-76.24951482387621	-145871.17638292973	-8262.167090898343	7.835333243563387e+16
9	483.50070707558143	924973.9481334805	52390.70947524073	3.1504828556250286e+18
10	-3065.892374537024	-5865287.3678424135	-332211.0106301894	1.266767081252219e+20

Table 9: For learning rate $\eta = 10^{-6}$

In the case of $\eta = 10^{-6}$, the objective loss value still diverges!

When $\eta = 10^{-7}$, the objective value converges!
For $\eta = 10^{-7}$:

iteration	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	objective value
0	0	0	0	11767.655172413792
1	2.1437931034482757e-05	0.04096499310344827	0.002379048275862069	1110.1129161163483
2	2.715513077533888e-05	0.05185502475457021	0.0030546406869051603	356.5798665121973
3	2.8692234240510285e-05	0.05474821527306015	0.0032772862594323427	303.272176071674
4	2.911786138375811e-05	0.055515070090112376	0.0033794915623561357	299.47126470347644
5	2.924794897171666e-05	0.05571653884810825	0.0034496690871070436	299.17054669283004
6	2.929945212342864e-05	0.055767673336805046	0.003511327448026648	299.117301133518
7	2.933005854909655e-05	0.05577883457201593	0.0035707175148007343	299.08155674735656
8	2.9355107236519346e-05	0.05577936721411364	0.00362950137499661	299.0470542879208
9	2.9378676776266748e-05	0.05577707391954384	0.0036881209728922364	299.0126442062925
10	2.9401851653335397e-05	0.0557740293910486	0.0037466938203544993	298.9782452273274

Table 10: For learning rate $\eta = 10^{-7}$

For the case $\eta = 10^{-7}$, even though the objective function converges, the value of β is still quite different from the closed form solution from Question 3.
As we decrease the value of η from 10^{-1}, the value of objective function keeps on diverging. Until $\eta \approx 10^{-7}$, the value of objective function converges along the iteration at the fastest convergence rate. However, as η keeps on decreasing, the value of objective function is still converging but converges at a much slower rate.
e.g. for $\eta = 10^{-8}$:

iteration	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	objective value
0	0	0	0	11767.655172413792
1	2.143793103448276e-06	0.004096499310344827	0.00023790482758620688	10145.667858626419
2	4.130378893960285e-06	0.00789224900616639	0.000458775096524224	8753.080208872163
3	5.971298006553971e-06	0.011409327207784387	0.0006638613158403545	7557.447877169021
4	7.677243874740856e-06	0.014668191274029657	0.0008543221943151623	6530.915171101077
5	9.258124923844816e-06	0.017687796782835195	0.0010312313795672714	5649.566071513553
6	1.0723122198690515e-05	0.02048570777742597	0.0011955837024189855	4892.86703848679
7	1.2080742760825796e-05	0.023078198919302635	0.0013483009628611389	4243.188622115123
8	1.3338869165837849e-05	0.0254803501421438	0.001490237291268519	3685.3947326063235
9	1.4504805308524616e-05	0.02770613435713672	0.001622184116046842	3206.4900005402114
10	1.558531890255832e-05	0.029768498719833142	0.0017448747666032637	2795.317011506272

Table 11: For learning rate $\eta = 10^{-8}$

In this case $\eta = 10^{-8}$, the objective loss value still converges. However, at a slower rate.
□

7. As preprocessing, normalize your year and Monona features (but not $y_{Mendota}$). Then repeat Q6.
For $\eta = 10^0$:

iteration	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	objective value
0	0	0	0	11767.655172413792
1	214.3793103448276	-10.422135611212198	29.46741047714559	12020.91633487461
2	-5.684341886080802e-14	24.928706899838758	-8.372366523033854	13137.591111272877
3	214.37931034482756	-42.054908670744325	58.93651580976627	16901.507763962
4	0.0	81.12593035494596	-64.28683074959588	29573.219800978753
5	214.3793103448276	-144.92644640862522	161.77110111713625	72233.91869340594
6	0.0	269.8464435131663	-253.0025147741125	215855.80410604266
7	214.37931034482787	-491.1927866010038	508.03681021348126	699374.5110612202
8	-6.536993168992922e-13	905.1865009863	-888.3424897723698	2327192.8571560695
9	214.37931034482938	-1656.9350465037073	1673.7790593379427	7807420.467312624
10	-3.609557097661309e-12	3044.1278074972493	-3027.283794874765	26257203.10501998

Table 12: For learning rate $\eta = 10^0$

For the case of $\eta = 10^0$, the objective loss value is diverging!

For $\eta = 10^{-1}$:

iteration	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	objective value
0	0	0	0	11767.655172413792
1	21.43793103448276	-1.0422135611212198	2.9467410477145592	7545.998373501811
2	38.588275862068976	-1.6266973410198067	5.220410220655868	4850.273518530399
3	52.30855172413794	-1.9015587758996588	6.993463365634234	3127.473895121937
4	63.28477241379311	-1.970844667076492	8.391542655956412	2025.6079196795692
5	72.06574896551724	-1.907266210297299	9.506512902731407	1320.3620154290259
6	79.09053020689655	-1.7612901693919225	10.405828814316633	868.6443341670564
7	84.7103552	-1.5676293302993598	11.139270319528144	579.0974806545676
8	89.20621519448275	-1.3498721557003726	11.74378940446234	393.3511885316241
9	92.80290319006896	-1.1237815047765827	12.24700145797667	274.08708818677417
10	95.68025358653793	-0.8996418081774248	12.669703292474024	197.43183896910952

Table 13: For learning rate $\eta = 10^{-1}$

Here, for the case of $\eta = 0.1$, we find the objective loss value already converges. As we keep on decreasing

the value of η, we can find the objective loss value is still converges but converges at a much slower rate. For example,
$\eta = 10^{-2}$:

iteration	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	objective value
0	0	0	0	11767.655172413792
1	2.143793103448276	-0.10422135611212198	0.29467410477145595	11303.019663371015
2	4.244710344827585	-0.2038654144120171	0.5826174907951793	10856.880645469235
3	6.303609241379309	-0.29908028233589046	0.8640026139179805	10428.499209186446
4	8.321330159999999	-0.3900096612564558	1.1389972486193198	10017.166089426179
5	10.298696660248275	-0.47679297373323376	1.4077646183744192	9622.200469635936
6	12.236515830491586	-0.5595654871289852	1.6704635223483257	9242.948834521754
7	14.13557861733003	-0.6384584336956088	1.9272484585246303	8878.783869365008
8	15.996660148431705	-0.7135991272298753	2.178269743369625	8529.103404030397
9	17.820520048911348	-0.7851110763965352	2.4236736281298135	8193.329399833321
10	19.607902751381395	-0.8531140948135592	2.6636024118579273	7870.906977510949

Table 14: For learning rate $\eta = 10^{-2}$

The objective loss function still converges but at a slower rate.
□

8. "Reset" to Q3 (no normalization, use closed-form solution), but train a linear regression model without using Monona:

$$\hat{y}_{Mendota} = \gamma_0 + \gamma_1 x.$$

(a) Interpret the sign of γ_1.
Explanation:

γ_0	4.06111060e+02
γ_1	-1.56298774e-01

Table 15: Result of γ_0 and γ_1

γ_1 is negative number. The meaning for this negative sign is to show that the ice days are decreasing, as years go on. I.e. $\widehat{y_{Mendota}}$ decreases as x increases.
□

(b) Some analysts claim that because β_1 the closed-form solution in Q3 is positive, fixing all other factors, as the years go by the number of Mendota ice days will increase, namely the model in Q3 indicates a cooling trend. Discuss this viewpoint, relate it to question 8(a).
Explanation:
Firstly, it is a correct way to describe the coefficient of the regressor "year", as fixing all other factors in this model. However, it is unreasonable for real life. Because the other regressor y_{Monona} is correlated with the regressor "year". So it is unreasonable and impractical in real life to change the value of "year", while holding the regressor "y_{Monona}" fixed. Even though the regressor "year" is positive while holding other factors as constant, it is not practical for real life case.

Secondly, there is always error existing in the model, data or experiment. It is likely to have other **omitted** regressors existing, which makes the coefficient of "year" to be **biased** in this model. Also, due to the existence of error in the model, we usually have a **confidence interval**. And we have to run **Hypothesis Test** on this model. Even though the coefficient of "year" is positive, it is likely to be not against our hypothesis that the coefficient is supposed to be negative, which can be depending on our tolerance of error or **Significance level**.

Thirdly, the regression model in $8(a)$ has not included the regressor "y_{Monona}". Therefore, the coefficient of $year$ in question $8(a)$ actually means the combining effect of changing of year and other

variables that are correlated with variable "year". Therefore, the coefficient of "year" in question $8(a)$ and $8(b)$ actually share different meaning. If we want to compare these two regressors whether equivalent to each other, we should run **F - Test**.
□

9. Of course, Weka has linear regression. Reset to Q3. Save the training data in .arff format for Weka. Use classifiers / functions / LinearRegression. Choose "Use training set." Bring up Linear Regression options, set "ridge" to 0 so it does not regularize. Run it and tell us the model: it is in the output in the form of "β_1 * year + β_2 * Monona + β_0."

```
=== Classifier model (full training set) ===


Linear Regression Model

y_mendota =

      0.0412 * year +
      0.853  * monona +
    -64.1828
```

Figure 3: Weka Output with Ridge = 0

10. Ridge regression.

    (a) Then set ridge to 1 and tell us the resulting Weka model.

```
=== Classifier model (full training set) ===


Linear Regression Model

y_mendota =

      0.0387 * year +
      0.8436 * monona +
    -58.3961
```

Figure 4: Weka Output with Ridge = 1

    (b) Meanwhile, derive the closed-form solution in matrix form for the ridge regression problem:

$$\min_{\beta} \left(\frac{1}{n} \sum_{i=1}^{n} (x_i^\top \beta - y_i)^2 \right) + \lambda \|\beta\|_A^2$$

where

$$\|\beta\|_A^2 := \beta^\top A\beta$$

and

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

This A matrix has the effect of NOT regularizing the bias β_0, which is standard practice in ridge regression. Note: Derive the closed-form solution, do not blindly copy lecture notes.

Solution:

Notation: we are using the same definition of **design matrix** X and Y in question $Q3$.

Therefore, we can re-write the objective funtion as:

$$\min_\beta \left(\frac{1}{n}||X\beta - Y||_2^2\right) + \lambda\|\beta\|_A^2 = \min_\beta \left(\frac{1}{n}||X\beta - Y||_2^2\right) + \lambda\beta^\top A\beta$$

Then, we can get the gradient as:

$$\nabla_\beta \left(\frac{1}{n}||X\beta - Y||_2^2\right) + \lambda\beta^\top A\beta = \frac{2}{n}X^\top(X\beta - Y) + \lambda\cdot(A + A^\top)\beta = 0$$

Since A is a **Symmetric Matrix**, we have $A^\top = A$, then:

$$\nabla_\beta \left(\frac{1}{n}||X\beta - Y||_2^2\right) + \lambda\beta^\top A\beta = \frac{2}{n}X^\top(X\beta - Y) + 2\cdot\lambda A\beta = 0$$

Then, let's solve for β:

$$(\frac{1}{n}\cdot I_{3\times3}\cdot X^\top X + \lambda\cdot I_{3\times3}\cdot A)\beta = \frac{1}{n}X^\top Y$$

$$\hat{\beta} = \frac{1}{n}(\frac{1}{n}\cdot I_{3\times3}\cdot X^\top X + \lambda\cdot I_{3\times3}\cdot A)^{-1}X^\top Y$$

Here, the part

$$\frac{1}{n}\cdot I_{3\times3}\cdot X^\top X + \lambda\cdot I_{3\times3}\cdot A$$

is invertible, because the front part is **positive semi-definite** and later part is **positive definite**, with $\lambda > 0$. Then the whole thing is **positive definite**.

□

(c) Let $\lambda = 1$ and tell us the value of β from your ridge regression model.

As $\lambda = 1$:

β_0	-6.23294723e+01
β_1	4.04390872e-02
β_2	8.49714502e-01

Table 16: value of β_0, β_1, β_1, with $\lambda = 1$ in ridge regression model

where, the coefficient is for the model $y_{Mendota} = \beta_0 + \beta_1\cdot x + \beta_2\cdot y_{Monona}$.

□