HOMEWORK 5

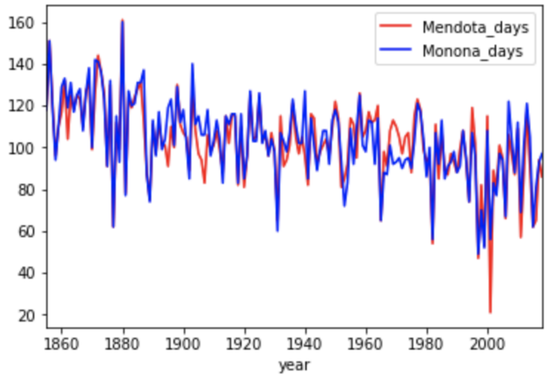
>>Haley Massa<<
>>9071903141<<

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.

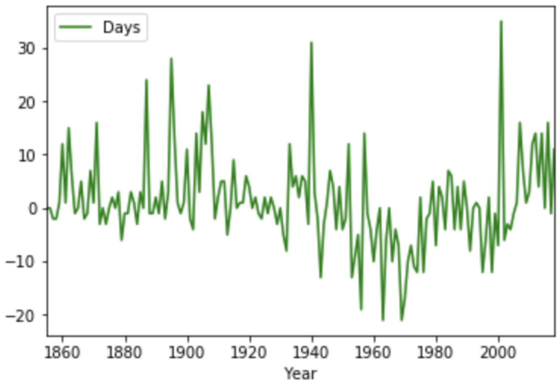
Linear Regression (100 pts total, 10 each)

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. $y_{Monona} - y_{Mendota}$. Below is the figure of Mendota and Monona of the year on the x axis and ice days on the y



Here is the plot of Monona - Mendota :



2. Split the datasets: $x \le 1970$ 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. The training mean for lake Mendota was 107.189 and for lake Monona was 108.4827. The standard deviation for lake Mendota was 16.74662 and for lake Monona was 18.122522.

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 $\beta_0, \beta_1, \beta_2$.

We need to create an X matrix that is: $\begin{bmatrix} 1 & ... & 1 \\ year(1) & ... & year(n) \\ yMonona(1) & ... & yMonona(n) \end{bmatrix}$

Where this matrix consists of the constant 1, all the year values and then Monona values.

We will also have a β matrix that will consist of

$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}$

With a y matrix that will consist of the true values of lake mendota:

$\begin{bmatrix} yMendota(1) \\ ... \\ yMentoda(n) \end{bmatrix}$

The MLE formula in matrix form is then:

$$= min(X\beta - y)^T(X\beta - y)$$

And solving for the betas is by setting this closed form to zero and solving for betas

$$= (X\beta - y)^T * (X\beta - y)$$
$$(X^T\beta^T - y^T)(X\beta - y)$$
$$(\beta^T X^T X\beta - \beta^T X^T y - y^T X\beta + y^T y)$$

$y^T y$ is a relative constant compared to beta $\beta^T X^T y = y^T X\beta$ since they are 1x1 transposes of each other

$$
\begin{aligned}
&(2(\frac{1}{2}\beta^TX^TX\beta - \beta^TX^Ty) + y^Ty) \\
&= 2X^TX\beta - 2X^Ty \\
&= 2X^T(X\beta - y) \\
&= 2X^TX\beta - 2X^Ty \\
&X^Ty = X^TX\beta
\end{aligned}
$$

when solving for beta:

$$\beta = (X^TX)^{-1}X^Ty$$

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.)

   The mean squared error is 124.26409. The R^2 value is 0.71049.

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.

$$
\begin{aligned}
&= \frac{\partial}{\partial\beta}\frac{1}{n}\sum_{i=1}^{n}\left(x_i^T\beta - y_i\right)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\beta}\left(x_i^T\beta - y_i\right)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}2\left(x_i^T\beta - y_i\right)(-x_i) \\
&= \frac{-2}{n}\sum_{i=1}^{n}\left(x_i^T\beta - y_i\right)(x_i)
\end{aligned}
$$

You can also derive the gradient using the matrix form the the equation, as seen above. I am pulling out a 1/n because it is seen in the sum above above so, that will appear at the very end.

$$
\begin{aligned}
&\bigtriangledown\beta(X\beta - y)^T(X\beta - y) \\
&\bigtriangledown\beta(X^T\beta^T - y^T)(X\beta - y) \\
&\bigtriangledown\beta(\beta^TX^TX\beta - \beta^TX^Ty - y^TX\beta + y^Ty)
\end{aligned}
$$

y^Ty is a relative constant compared to beta $\beta^TX^Ty = y^TX\beta$ since they are 1x1 transposes of each other

$$
\begin{aligned}
&\bigtriangledown\beta(2(\frac{1}{2}\beta^TX^TX\beta - \beta^TX^Ty) + y^Ty) \\
&= 2X^TX\beta - 2X^Ty \\
&= 2X^T(X\beta - y) \\
&= 2X^TX\beta - 2X^Ty) \\
&= \frac{2}{n}(X^TX\beta - x^Ty)
\end{aligned}
$$

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.

Table 1: Gradient Descent with no normalization

Step Number	Beta_0	Beta_1	Beta_2
1	2.14379310e+01	4.09649931e+04	2.37904828e+03
2	-1.57206884e+07	-3.00748795e+10	-1.70345111e+09
3	1.15405879e+13	2.20780301e+16	1.25050392e+15
4	-8.47196890e+18	-1.62075266e+22	-9.17997455e+20
5	6.21928952e+24	1.18979781e+28	6.73903790e+26
6	-4.56559302e+30	-8.73432978e+33	-4.94714136e+32
7	3.35161108e+36	6.41188917e+39	3.63170649e+38
8	-2.46042448e+42	-4.70698081e+45	-2.66604308e+44
9	1.80620259e+48	3.45540414e+51	1.95714762e+50
10	-1.32593697e+54	-2.53661917e+57	-1.43674603e+56

That was a table of the beta values, here is a table of the objective function values. Calculated by

$$\frac{1}{n}\sum_{i=1}^{n}(x_i^T\beta_{stepnumber} - y_i)^2$$

Table 2: Gradient Descent with no normalization Objective Function Values

Step Number	Objective Function Value (MSE)
1	6180350808297760.0
2	3.330626781514358e+27
3	1.794894028506587e+39
4	9.672787691041722e+50
5	5.212721209720432e+62
6	2.809165597156087e+74
7	1.5138755814390718e+86
8	8.158363032772642e+97
9	4.396589005765002e+109
10	2.3693472339933764e+121

It was getting very large, so I do not believe it is converging.

The real beta values for the closed form solution are:

[-6.41827663e+01, 4.12245664e-02, 8.52950638e-01])

7. As preprocessing, normalize your year and Monona features (but not $y_{Mendota}$). Then repeat Q6.

Table 3: Gradient Descent with normalization of Monona features and year

Step Number	Beta_0	Beta_1	Beta_2
1	21.43793103	-1.04221356	2.94674105
2	38.58827586	-1.62669734	5.22041022
3	52.30855172	-1.90155878	6.99346337
4	63.28477241	-1.97084467	8.39154266
5	72.06574897	-1.90726621	9.5065129
6	79.09053021	-1.76129017	10.40582881
7	84.7103552	-1.56762933	11.13927032
8	89.20621519	-1.34987216	11.7437894
9	92.80290319	-1.1237815	12.24700146
10	95.68025359	-0.89964181	12.66970329

That was a table of the beta values, here is a table of the objective function values. Calculated by

$$\frac{1}{n}\sum_{i=1}^{n}(x_i^T\beta_{stepnumber}-y_i)^2$$

Table 4: Gradient Descent with normalization Objective Function Values

Step Number	Objective Function Value (MSE)
1	7545.998373501811
2	4850.273518530398
3	3127.4738951219365
4	2025.6079196795708
5	1320.362015429026
6	868.6443341670561
7	579.097480654567
8	393.35118853162385
9	274.0870881867739
10	197.4318389691092

We can see that with normalization it is getting closer to converging, but I don't believe it has converged yet.

The closed form beta values are : ([107.18965517, 1.38639633, 15.45761632])

Which we have not gotten to in the gradient descent, but they are getting closer.

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. We get the following equation:

$$\hat{y}_Mendota = 406.1106 + -0.15629877 * x$$

This shoes that the sign of the γ_1 is negative. This is saying that every year that number of days will decrease. Whihc makes sense, there is a cooling affect throughout the years.

(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).

I think that viewpoint is wrong. Since the model in Q3 takes into account the Monona lake days, the positive value in that beta takes different factors into account. We know that other factors matter because in 8A, the value is negative, but it doesn't take into account the Monona data. It is predicting the exact same thing as Q3, so they are predicting overall the same trends. So just picking out one feature to indicate a viewpoint overall might not be the best method for those extras.

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."

$$= 0.0412 * year + 0.853 * Monona - 64.1828$$

10. Ridge regression.

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

$$= -58.3961 + 0.0387 * year + 0.8442 * Monona$$

(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. In matrix form:

$$= (X\beta - y)^T (X\beta - y) + \lambda \beta^T A \beta$$

Now we need to find the minimum in regards to beta and we do that through taking the derivative

$$\frac{\partial}{\partial \beta} (X\beta - y)^T (X\beta - y) + \frac{\partial}{\partial \beta} \lambda \beta^T A \beta$$

We know from the derivation in problem five that the first part equals

$$\frac{\partial}{\partial \beta} (X\beta - y)^T (X\beta - y) = 2X^T X\beta - 2X^T y$$

The second part equals

$$\frac{\partial}{\partial \beta} \lambda \beta^T A \beta = 2\lambda A \beta$$

We once again have a bunch of factors of 2, when we cancel them out we get

$$= X^T X\beta - X^T y + \lambda A \beta$$

We set this equal to zero to solve for the closed form solution

$$0 = X^T X\beta - X^T y + \lambda A \beta$$

$$X^T y = X^T X\beta + \lambda A \beta$$

$$X^T y = (X^T X + \lambda A)\beta$$

Then solving for beta

$$\beta = (X^T X + \lambda A)^{-1} X^T y$$

(c) Let $\lambda = 1$ and tell us the value of β from your ridge regression model. We get $\beta = [-62.32947, 0.0404390, 0.849714502])$