

Name: Harris Spahic

Pledge: "I pledge my honor I've
abided by the Stevens Honor System"

- Harris Spahic

$$\begin{aligned}
 \#1) * P(CS) &= P(S_1) \times P(CS|S_1) \\
 &\quad + P(S_2) \times P(CS|S_2) \\
 &\quad + P(S_3) \times P(CS|S_3) \\
 &= 0.2 \times \frac{6}{20} \\
 &\quad + 0.2 \times \frac{10}{20} \\
 &\quad + 0.6 \times \frac{6}{20} \\
 &= 0.8 \left(\frac{6}{20} \right) + 0.2 \left(\frac{10}{20} \right) \\
 &= \frac{4.8}{20} + \frac{2}{20} = \frac{6.8}{20} = \frac{34}{100} = 0.34
 \end{aligned}$$

$$\begin{aligned}
 1.2) P(S_3 | \text{Stat}) &= \frac{P(S_3) \times P(\text{Stat}|S_3)}{P(\text{Stat})} \\
 &= \frac{0.6 \left(\frac{6}{20} \right)}{\frac{24}{60}} = \frac{\frac{3.6}{24} \times 60}{\frac{10.8}{24}} = \frac{10.8}{24} = 0.45
 \end{aligned}$$

#2) Since the weights are normally distributed, the Likelihood function is

Probability Mass Function

$$= \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right)$$

$$\begin{aligned}
 \rightarrow LE(\mu, \sigma^2) &= \prod_{i=1}^n f(x_i; \mu, \sigma^2) \\
 &= \left(\frac{1}{\sigma \sqrt{2\pi}} \right)^n \exp\left(\sum_{i=1}^n -\frac{(x_i - \mu)^2}{2\sigma^2} \right) \\
 &= \left(\frac{1}{\sigma} \right)^n \cdot \left(\frac{1}{2\pi} \right)^{\frac{n}{2}} \exp\left(\frac{1}{2\sigma^2} \sum_{i=1}^n -(x_i - \mu)^2 \right)
 \end{aligned}$$

Why can't this be "minimum LE"

2.2) To find the $\text{MLE}(\mu)$ & $\text{MLE}(\sigma^2)$ we must take $\frac{\partial}{\partial \theta}$, where θ is the parameter in question, and set it equal to 0.

Looking at our $\text{LE}(\mu, \sigma)$ from #2, $\frac{\partial}{\partial \theta}$ is hard to compute. But notice $\ln(x)$ is a continuously increasing function, hence the max of $\ln(\text{MLE}(\theta))$ and $\text{max } \text{MLE}(\theta)$ both at same θ .

$$\rightarrow \text{MLE}(\mu) \Rightarrow \frac{\partial}{\partial \mu} \ln(\text{LE}(\mu, \sigma^2)) = 0$$

$$= \frac{\partial}{\partial \mu} \left(-n \ln(\sigma) - \frac{n}{2} \ln(2\pi) + \left(\frac{1}{2\sigma^2} \left(\sum_{i=1}^n -(x_i - \mu)^2 \right) \right) \right)$$

$$= 0 + 0 + \frac{1}{2\sigma^2} \left(\sum_{i=1}^n -2(x_i - \mu) \cdot -1 \right)$$

$$= \frac{1}{2\sigma^2} \left(\sum_{i=1}^n x_i - n\mu \right) = 0$$

$$\Rightarrow \sum_{i=1}^n x_i - n\mu = 0$$

$$\Rightarrow \frac{\sum_{i=1}^n x_i}{n} = \mu_{ML}$$

$$\rightarrow \text{MLE}(\sigma^2) \Rightarrow (\text{MLE}(\theta_2)) \text{ where } \theta_2 = \sigma^2$$

$$\rightarrow \text{LE}(\mu, \theta_2) = \left(\frac{1}{\theta_2} \right)^n \cdot \left(\frac{1}{2\pi} \right)^{\frac{n}{2}} \exp \left(\frac{1}{2\theta_2} \sum_{i=1}^n -(x_i - \mu)^2 \right)$$

$$\rightarrow \text{MLE}(\theta_2) = \frac{\partial}{\partial \theta_2} \ln(\text{LE}(\mu, \theta_2))$$

$$\begin{aligned}
 &= \frac{\partial}{\partial \theta_2} \left(-\frac{n}{2} \ln(\theta) - \frac{n}{2} (2\pi) + \left(\frac{1}{2\theta_2} \sum_{i=1}^n (x_i - \mu)^2 \right) \right) \\
 &= -\frac{n}{2\theta_2} - 0 + \frac{1}{2\theta_2^2} \cdot \sum_{i=1}^n (x_i - \mu)^2 = 0 \\
 \Rightarrow & -n\theta_2 + \sum_{i=1}^n (x_i - \mu)^2 = 0 \quad \text{Since } \mu = \bar{x} \text{ by MLE}(\mu) \\
 \Rightarrow & \sum_{i=1}^n (x_i - \mu)^2 = n\theta_2 \\
 \Rightarrow & \frac{\sum_{i=1}^n (x_i - \mu_{ML})^2}{n} = \theta_2 = \sigma_{ML}^2
 \end{aligned}$$

Thus according to $MLE(\mu)$.

$$\hat{\mu} = \frac{112 + 120 + 131 + 126 + 145 + 158 + 157 + 136 + 148 + 176}{10}$$

$$= \frac{1409}{10} = 140.9$$

$$\begin{aligned}
 \hat{\sigma}^2 &= \frac{(112 - 140.9)^2 + \dots + (176 - 140.9)^2}{10} \\
 &= 346.69
 \end{aligned}$$

$$3.1) L.E(q) = \prod_{i=1}^n f(x_i; q)$$

where $f(x) = \frac{\text{Probability}}{\text{Mass Function}}$

Thus,

$$L.E(\mu, \sigma^2) = f(x_1; q)^{n_1} \\ f(x_2; q)^{n_2} \\ f(x_3; q)^{n_3} \\ f(x_4; q)^{n_4}$$

where n_x are the # of occurrences of x_i in our sample.

$$= \left(\frac{2q}{3}\right)^{n_1} \left(\frac{q}{3}\right)^{n_2} \left(\frac{2(1-q)}{3}\right)^{n_3} \left(\frac{1-q}{3}\right)^{n_4}$$

Our sample

$$(1, 1, 2, 2, 2, 3, 3, 3, 4, 4)$$

$$\rightarrow \left(\frac{2q}{3}\right)^2 \left(\frac{q}{3}\right)^3 \left(\frac{2(1-q)}{3}\right)^3 \left(\frac{1-q}{3}\right)^2$$

$$= \frac{2^5 q^5 (1-q)^5}{3^{10}}$$

3.2) $LE(q) = \left(\frac{2q}{3}\right)^{n_1} \left(\frac{q}{3}\right)^{n_2} \left(\frac{2(1-q)}{3}\right)^{n_3} \left(\frac{1-q}{3}\right)^{n_4}$

\rightarrow To find $MLE(q_b)$,

$\frac{\partial}{\partial q} LE(q) = 0$ since $\ln(x)$ is an continuously increasing function, the value of q_b that maximizes $LE(q)$ is the same as $\ln(LE(q_b))$.

$$\rightarrow \frac{\partial}{\partial q} \ln \left(\left(\frac{2q}{3}\right)^{n_1} \left(\frac{q}{3}\right)^{n_2} \left(\frac{2(1-q)}{3}\right)^{n_3} \left(\frac{1-q}{3}\right)^{n_4} \right) = 0$$

$$\rightarrow \frac{\partial}{\partial q} \left(n_1 \ln \left(\frac{2q}{3}\right) + n_2 \ln \left(\frac{q}{3}\right) + n_3 \ln \left(\frac{2(1-q)}{3}\right) + n_4 \ln \left(\frac{1-q}{3}\right) \right) = 0$$

All constants will come out of $\ln(\sim)$
 & be eliminated by d/dq

$$\rightarrow \frac{d}{dq} (n_1 \ln(q_b) + n_2 \ln(q_b) + n_3 \ln(1-q_b) + n_4 \ln(1-q_b))$$

$$= \frac{n_1}{q_b} + \frac{n_2}{q_b} - \frac{n_3}{(1-q_b)} - \frac{n_4}{(1-q_b)} = \frac{n_1+n_2}{q_b} - \frac{n_3+n_4}{(1-q_b)}$$

$$= \frac{n_1+n_2 - q_b(n_1+n_2+n_3+n_4)}{q_b(1-q_b)} = 0$$

$$\rightarrow \frac{n_1+n_2}{(n_1+n_2+n_3+n_4)} = q_{bML}$$

Since $n_1 = 2$

$$n_2 = 3$$

$$n_3 = 3$$

$$n_4 = 2$$

$q_{bML} = \frac{1}{2}$

Notice $q_b = 0$ & $q_b = 1$, then

$$LE(q_b) = 0.$$

Thus $q = \frac{1}{2}$ is the most likely value of q given the sample.

4) Recall:

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)}$$

which implies

$$P(\theta|x) \propto P(x|\theta)P(\theta)$$

Notice: $P(x|\theta)$ is equivalent to the likelihood function.

Thus finding the maximum posterior function ($P(\theta|x)$), is equivalent to maximizing the likelihood $\times P(\theta)$.

Then given

N input values, $x = (x_1, x_2, \dots, x_n)^T$

and target values $y = (y_1, \dots, y_N)^T$;

we estimate the target using $f(x, w)$. Assuming target variables drawn from a Gaussian Distribution

$$P(y|x, w, \beta) = N(y|f(x, w), \beta^{-1})$$

& a prior Guassian Distribution for w

$$P(w|\alpha) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left(-\frac{\alpha}{2} w^T w\right)$$

We proceed to find the MAP using
 $\max(P(y|x, w, \beta) \times P(w|\alpha))$

$$P(w|x, y, \alpha, \beta) \propto P(y|x, w, \beta) P(w|\alpha)$$

$$P(w|x, y, \alpha, \beta) \propto \prod_{n=1}^N N(y|f(x_n, w), \beta^{-1}) P(w|\alpha)$$

$$\rightarrow \underbrace{\left((\beta)^{N/2} \left(\frac{1}{2\pi}\right)^{N/2} \exp\left(-\frac{1}{2\beta} \sum_{n=1}^N ((f(x_n, w) - y_n)^2)\right) \right)}_{\text{constant}} \left(\frac{\alpha}{2\pi} \right)^{(M+1)/2} \exp\left(-\frac{\alpha}{2} w^T w\right)$$

maximize log in terms w

$$\frac{\partial}{\partial w} \left(\frac{N}{2} \ln \beta - \frac{N}{2} \ln 2\pi - \frac{\beta}{2} \sum_{n=1}^N (f(x_n, w) - y_n)^2 + \frac{M+1}{2} \ln \left(\frac{\alpha}{2\pi}\right) - \frac{\alpha}{2} w^T w \right)$$

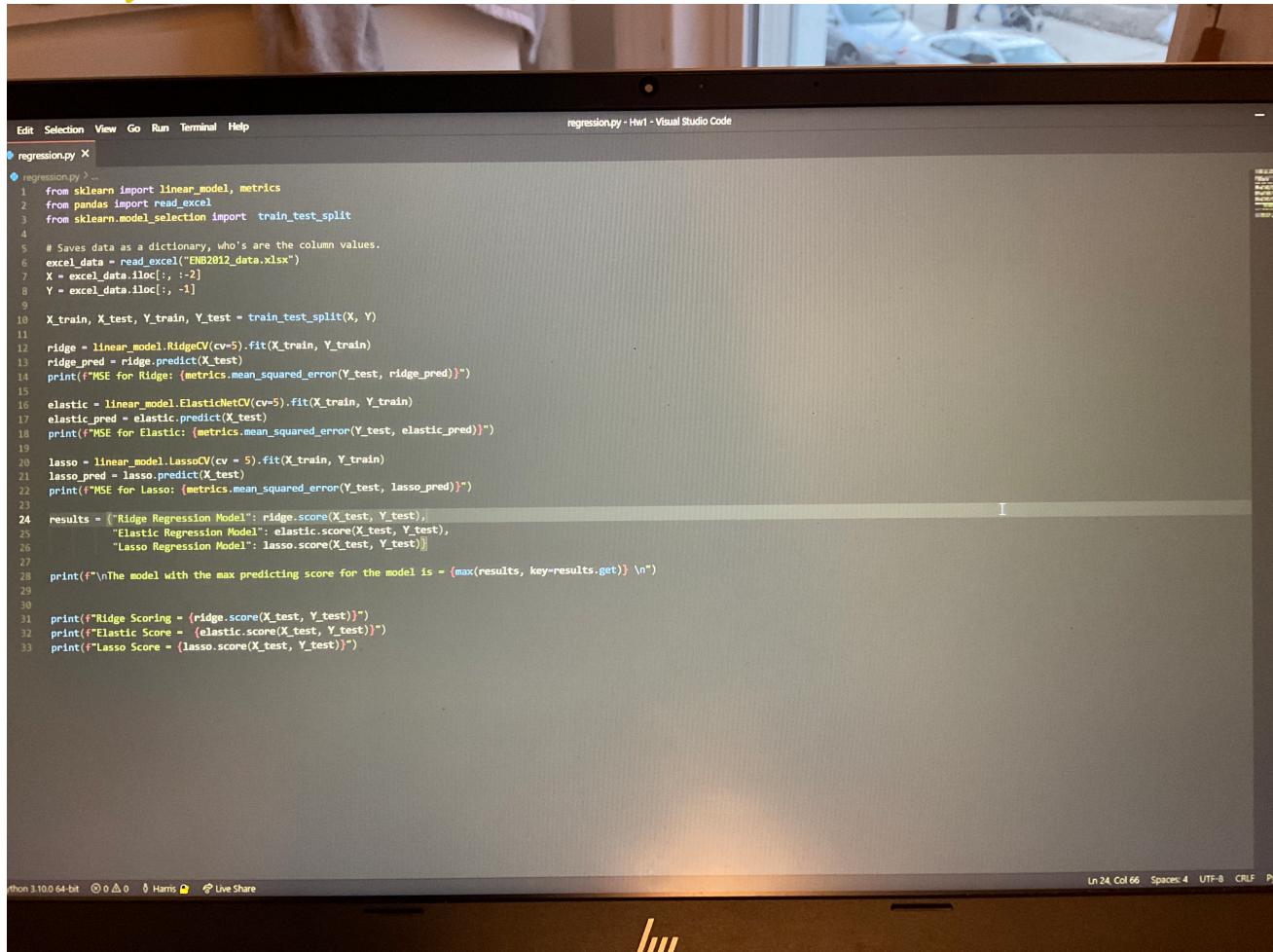
$$= -\frac{\beta}{2} \sum_{n=1}^N (f(x_n, w) - y_n)^2 - \frac{\alpha}{2} w w^T$$

Notice maximizing log in w is equivalent to minimizing the negative log in w .

$$\rightarrow = \frac{\partial}{\partial w} \left(\frac{\beta}{2} \sum_{n=1}^N (f(x_n, w) - y_n)^2 \right) + \frac{\alpha}{2} w w^T = 0$$

Or the minimizing of the negative sum-of-squares error function.

#5) Source Code



```

Edit Selection View Go Run Terminal Help
regression.py X
regression.py > ...
1  from sklearn import linear_model, metrics
2  from pandas import read_excel
3  from sklearn.model_selection import train_test_split
4
5  # Saves data as a dictionary, who's are the column values.
6  excel_data = read_excel("ENB2012_data.xlsx")
7  X = excel_data.iloc[:, :-2]
8  Y = excel_data.iloc[:, -1]
9
10 X_train, X_test, Y_train, Y_test = train_test_split(X, Y)
11
12 ridge = linear_model.RidgeCV(cv=5).fit(X_train, Y_train)
13 ridge_pred = ridge.predict(X_test)
14 print(f"MSE for Ridge: {metrics.mean_squared_error(Y_test, ridge_pred)}")
15
16 elastic = linear_model.ElasticNetCV(cv=5).fit(X_train, Y_train)
17 elastic_pred = elastic.predict(X_test)
18 print(f"MSE for Elastic: {metrics.mean_squared_error(Y_test, elastic_pred)}")
19
20 lasso = linear_model.LassoCV(cv = 5).fit(X_train, Y_train)
21 lasso_pred = lasso.predict(X_test)
22 print(f"MSE for Lasso: {metrics.mean_squared_error(Y_test, lasso_pred)}")
23
24 results = [{"Ridge Regression Model": ridge.score(X_test, Y_test),
25             "Elastic Regression Model": elastic.score(X_test, Y_test),
26             "Lasso Regression Model": lasso.score(X_test, Y_test)}]
27
28 print(f"\nThe model with the max predicting score for the model is = {max(results, key=results.get)} \n")
29
30
31 print(f"Ridge Scoring = {ridge.score(X_test, Y_test)}")
32 print(f"Elastic Score = {elastic.score(X_test, Y_test)}")
33 print(f"Lasso Score = {lasso.score(X_test, Y_test)}")

```

Python 3.10.0 64-bit ① 0 △ 0 ⚡ Harris ⚡ Live Share

Ln 24, Col 66 Spaces: 4 UTF-8 CRLF Pg

Evaluation Result

```
λ python regression.py
MSE for Ridge: 11.277082258961377
MSE for Elastic: 18.81610889454032
MSE for Lasso: 16.38916106418985

The model with the max predicting score for the model is --> Ridge Regression Model
Ridge Scoring = 0.8731087122331425
Elastic Score = 0.7882785428391881
Lasso Score = 0.8155868946899447

C:\Users\Bikeh\Desktop\PrgmProjects\CS559\Hw1
λ python regression.py
MSE for Ridge: 8.889270191169627
MSE for Elastic: 17.21975035112787
MSE for Lasso: 14.55693962493675

The model with the max predicting score for the model is : Ridge Regression Model
Ridge Scoring = 0.899521211428377
Elastic Score = 0.8053586367611328
Lasso Score = 0.8354574933237484

C:\Users\Bikeh\Desktop\PrgmProjects\CS559\Hw1
λ python regression.py
MSE for Ridge: 11.58130935062243
MSE for Elastic: 19.930639687575773
MSE for Lasso: 17.45043368684944

The model with the max predicting score for the model is : Ridge Regression Model
Ridge Scoring = 0.8819035826952616
Elastic Score = 0.7966944024008691
Lasso Score = 0.821892123510543

C:\Users\Bikeh\Desktop\PrgmProjects\CS559\Hw1
λ python regression.py
MSE for Ridge: 11.831456296841855
MSE for Elastic: 20.886188844820786
MSE for Lasso: 17.956320203473673

The model with the max predicting score for the model is = Ridge Regression Model
Ridge Scoring = 0.865019388180432
Elastic Score = 0.76171736782686803
Lasso Score = 0.7951431314723043

C:\Users\Bikeh\Desktop\PrgmProjects\CS559\Hw1
λ |
```