2022S CS 559-B: Homework 1

NAME: JANMEJAY MOHANTY

Solution 1:

Given:

Probability of session 1, p(S1) = 0.2

Probability of session 2, p(S2) = 0.2

Probability of session 3, p(S3) = 0.6

Major	**Session 1**	**Session 2**	**Session 3**
CS	$6 * 0.2 = 1.2$	$10 * 0.2 = 2$	$6 * 0.6 = 3.6$
STAT	$8 * 0.2 = 1.6$	$10 * 0.2 = 2$	$6 * 0.6 = 3.6$
MG	$6 * 0.2 = 1.2$	$0 * 0.2 = 0$	$3 * 0.6 = 1.8$

Total Major CS = $1.2 + 2 + 3.6 = 6.8$

Total Major STAT = $1.6 + 2 + 3.6 = 7.2$

Total Major MG = $1.2 + 0 + 1.8 = 3$

(1) The probability of a selected student majored in CS:

Probability for CS student, p(CS) = $\frac{\text{Total Major CS}}{\text{Total Students}}$ = $\frac{6.8}{17}$ = 0.4

(2) The probability of selected student is from STAT comes from Session 3:

Probability for STAT student from Session 3, p(STAT3) = $\frac{\text{STAT in Session 3}}{\text{Total Major STAT}}$ =

$$\frac{3.6}{7.2} = 0.5$$

Solution 2:

1) The weights of the students are as follows:

Let x = {112, 120, 131, 126, 145, 158, 157, 136, 148, 176} (Given)

Taking consider of the student's weights as an independent event, then the likelihood function is given as follows:

$$P(x|\mu,\ \sigma^2) = \prod_{n=1}^{N} \mathcal{N}(x_n|\mu,\ \sigma^2) \qquad \text{-----(1)}$$

But, for a normal distribution, the pdf (probability density function) is given below:

$$\mathcal{N}(x|\mu,\ \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left[\frac{-1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]} \qquad \text{-----(2)}$$

Therefore,

Taking combining both equations (1) and (2) and also putting the values, we get:

$$P(x|\mu,\ \sigma^2) = \prod_{n=1}^{N=10} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left[\frac{-1}{2}\left(\frac{x_n-\mu}{\sigma}\right)^2\right]}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left[\frac{-1}{2}\left(\frac{x_1-\mu}{\sigma}\right)^2\right]} * \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left[\frac{-1}{2}\left(\frac{x_2-\mu}{\sigma}\right)^2\right]} * \dots\dots\dots * \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left[\frac{-1}{2}\left(\frac{x_{10}-\mu}{\sigma}\right)^2\right]}$$

$$P(x|\mu,\ \sigma^2) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{10} e^{\left[\frac{-1}{2}\left(\sum_{n=1}^{10}\left(\frac{x_n-\mu}{\sigma}\right)^2\right)\right]}$$

Here, the final expression,

$$\boldsymbol{P(x|\mu,\ \sigma^2) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{10} e^{\left[\frac{-1}{2}\left(\sum_{n=1}^{10}\left(\frac{x_n-\mu}{\sigma}\right)^2\right)\right]}} \qquad \text{------(3)}$$

Is the likelihood function for the given dataset

$x = \{112, 120, 131, 126, 145, 158, 157, 136, 148, 176\}$.

2) For maximizing the likelihood function and also to find both the μ and σ^2 values, we need to determine first the roots of the derivatives with respect to μ and σ.

First, we need to take logarithm throughout the likelihood function which is also known as log-likelihood.

From equation (3), we get:

$$P(x|\mu,\ \sigma^2) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{10} e^{\left[\frac{-1}{2}\left(\sum_{n=1}^{10}\left(\frac{x_n-\mu}{\sigma}\right)^2\right)\right]} \qquad \text{------(4)}$$

Now, taking the logarithm throughout the equation (4), we get:

$$\log P(x|\mu,\ \sigma^2) = \log\left[\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{10} e^{\left[\frac{-1}{2}\left(\sum_{n=1}^{10}\left(\frac{x_n-\mu}{\sigma}\right)^2\right)\right]}\right]$$

$$= \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{10} + \log\left[e^{\left[\frac{-1}{2}\left(\sum_{n=1}^{10}\left(\frac{x_n-\mu}{\sigma}\right)^2\right)\right]}\right]$$

$$= 10\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{1}{2\sigma^2}\sum_{n=1}^{10}(x_n-\mu)^2$$

$$= 10\log(1) - 10\log\left(\sigma\sqrt{2\pi}\right) - \frac{1}{2\sigma^2}\sum_{n=1}^{10}(x_n-\mu)^2$$

$$\log P(x|\mu,\ \sigma^2) = -10\log(\sigma) - 10\log\left(\sqrt{2\pi}\right) - \frac{1}{2\sigma^2}\sum_{n=1}^{10}(x_n-\mu)^2 \qquad \text{------(5)}$$

Now, taking equation (5) and doing its derivative with respect to mean μ, we get:

$$\frac{d\,[\log P(x|\mu,\ \sigma^2)]}{d\mu} = \frac{d\left[-10\log(\sigma) - 10\log\left(\sqrt{2\pi}\right) - \frac{1}{2\sigma^2}\sum_{n=1}^{10}(x_n-\mu)^2\right]}{d\mu}$$

$$= \frac{d\left[\frac{-1}{2\sigma^2} * 2\sum_{n=1}^{10}(x_n-\mu)\right]}{d\mu}$$

$$\frac{d\left[\log P\left(x|\mu,\ \sigma^2\right)\right]}{d\mu} = \frac{-1}{\sigma^2}\left[\sum_{n=1}^{10}(x_n-\mu)\right]$$

Now, let's consider the mean (μ), we try to find the root of $\frac{d\left[\log P\left(x|\mu,\ \sigma^2\right)\right]}{d\mu}$.

Therefore,

=> $\frac{-1}{\sigma^2}\left[\sum_{n=1}^{10}(x_n-\mu)\right] = 0$

As, $\sigma^2 > 0$

Then,

$\sum_{n=0}^{10}(x_n-\mu) = 0$

=> $(x_1-\mu) + (x_2-\mu) + \cdots + (x_{10}-\mu) = 0$

=> $\sum_{n=1}^{10}(x_n) - 10\mu = 0$

=> $-10\mu = -\sum_{n=1}^{10}(x_n)$

=> $\mu = \frac{1}{10}\sum_{n=1}^{10}(x_n)$

=> $\mu = \frac{1}{10}(112 + 120 + 131 + 126 + 145 + 158 + 157 + 136 + 148 + 176)$

=> $\mu = \frac{1}{10} * 1409$

=> $\boldsymbol{\mu = 140.9}$

Now, again we take the equation (5) and getting its derivatives with respect to σ,

$$\frac{d\,[\log P(x|\mu,\ \sigma^2)]}{d\sigma} = \frac{d\ \left[-10\log(\sigma) - 10\log(\sqrt{2\pi})\ - \frac{1}{2\sigma^2}\sum_{n=1}^{10}(x_n-\mu)^2\ \right]}{d\sigma}$$

$$= \frac{d[-10\log(\sigma)]}{d\sigma} - \frac{d\left[\frac{-1}{2\sigma^2}\sum_{n=1}^{10}(x_n-\mu)^2\right]}{d\sigma}$$

$$= -10 * \frac{1}{\sigma} + \frac{2}{2\sigma^3}\sum_{n=1}^{10}(x_n-\mu)^2$$

$$\frac{d\left[\log P\left(x\middle|\mu,\ \ \sigma^2\right)\right]}{d\sigma} = \frac{-10}{\sigma} + \frac{1}{\sigma^3}\sum_{n=1}^{10}(x_n-\mu)^2$$

Now, let's consider the σ, we try to find out the roots of $\frac{d\left[\log P\left(x\middle|\mu,\ \ \sigma^2\right)\right]}{d\sigma}$

$$=> \frac{-10}{\sigma} + \frac{1}{\sigma^3}\sum_{n=1}^{10}(x_n-\mu)^2 = 0$$

$$=> \frac{1}{\sigma}\left[-10 + \frac{1}{\sigma^2}\sum_{n=1}^{10}(x_n-\mu)^2\right] = 0$$

As, $\sigma > 0$

$=> \frac{1}{\sigma^2}\sum_{n=1}^{10}(x_n-\mu)^2 = 10$

$=> \sigma^2\ = \frac{\sum_{n=1}^{10}(x_n-\mu)^2}{10}$

$=> \sigma^2\ = \frac{1}{10}[(x_1-\mu)^2 + (x_2-\mu)^2 + \cdots + (x_{10}-\mu)^2]$

$$=> \sigma^2\ = \frac{1}{10}[(112-140.9)^2 + (120-140.9)^2 + (131-140.9)^2 + (126-140.9)^2 + (145-140.9)^2 + (145-140.9)^2 + (158-140.9)^2 + (157-140.9)^2 + (136-140.9)^2 + (148-140.9)^2 + (176-140.9)^2]$$

$$=> \sigma^2\ = \frac{1}{10}[(28.9)^2 + (20.9)^2 + (9.9)^2 + (14.9)^2 + (-4.1)^2 + (-17.1)^2 + (-16.1)^2 + (4.9)^2 + (-7.1)^2 + (-35.1)^2]$$

$=> \sigma^2\ = \frac{1}{10} * 3466.9$

$=> \boldsymbol{\sigma^2 = 346.69}$

Solution 3:

1) Let us take the frequency of X_i as f_i where $i = 1, 2, 3, 4$.

Suppose if we have n size of samples, then:

$$n = f_1 + f_2 + f_3 + f_4 \quad \text{-------(1)}$$

Now using the Likelihood function,

$$L(x, q) = Likelihood\ function\ for\ X_i, \ldots., X_n.$$

$$= \prod_{i=1}^{n} P[X_i = x_i]$$

$$= (P[X_1 = 1])^{f_1} * (P[X_2 = 2])^{f_2} * (P[X_3 = 3])^{f_3} * (P[X_4 = 4])^{f_4}$$

$$= \left(\frac{2q}{3}\right)^{f_1} * \left(\frac{q}{3}\right)^{f_2} * \left(\frac{2(1-q)}{3}\right)^{f_3} * \left(\frac{1-q}{3}\right)^{f_4}$$

$$= \left(\frac{2}{3}\right)^{f_1} * (q)^{f_1} * \left(\frac{q}{3}\right)^{f_2} * \left(\frac{2}{3}\right)^{f_3} * (1-q)^{f_3} * \left(\frac{1}{3}\right)^{f_4} * (1-q)^{f_4}$$

$$= \left(\frac{2}{3}\right)^{f_1} * \left(\frac{1}{3}\right)^{f_2} * \left(\frac{2}{3}\right)^{f_3} * \left(\frac{1}{3}\right)^{f_4} * (q)^{f_1} * (q)^{f_2} * (1-q)^{f_3} * (1-q)^{f_4}$$

$$= \frac{2^{f_1+f_3}}{3^{f_1+f_2+f_3+f_4}} * q^{f_1+f_2} * (1-q)^{f_3+f_4}$$

$$= \frac{2^{f_1+f_3}}{3^n} * q^{f_1+f_2} * (1-q)^{f_3+f_4} \quad \textbf{From equation (1)}$$

$$\boldsymbol{L(x, q) = \frac{2^{f_1+f_3}}{3^n} * q^{f_1+f_2} * (1-q)^{f_3+f_4}} \quad \textbf{-------(2)}$$

Taking logarithm throughout the equation (2),

$$\log(L(x, q)) = \log(2^{f_1+f_3}) + \log(3^n) + \log(q^{f_1+f_2}) + \log[(1-q)^{f_3+f_4}]$$

$$= (f_1 + f_3) * \log(2) - (n) * \log(3) + (f_1 + f_2) * \log(q) + (f_3 + f_4) * \log(1-q) \quad \text{---(3)}$$

Now, rearranging the dataset $\{1, 1, 2, 2, 2, 3, 3, 3, 4, 4\}$

Here,

$f_1 = 2$ (Number of 1's occurrence)

$f_2 = 3$ (Number of 2's occurrence)

$f_3 = 3$ (Number of 3's occurrence)

$f_4 = 2$ (Number of 4's occurrence)

=> $n = f_1 + f_2 + f_3 + f_4$ **From equation (1)**

=> $n = 2 + 3 + 3 + 2 = 10$

Putting values of n, f_1, f_2, f_3, f_4 in equation (3), we get:

$\log\big(L(x,q)\big) = (2+3)*\log(2) - (10)*\log(3) + (2+3)*\log(q) + (3+2)*\log(1-q)$
$\mathbf{\log\big(L(x,q)\big) = 5*\log(2) - 10*\log(3) + 5*\log(q) + 5*\log(1-q)}$ **-------(4)**

2) For getting the Maximum Likelihood Estimation (MLE) of q we need to solve the equation (4),

$$\frac{\partial\big(\log\big(L(x,q)\big)\big)}{\partial q} = 0$$

=> $\frac{\partial(\,5*\log(2)-10*\log(3)+5*\log(q)+5*\log(1-q))}{\partial q} = 0$

=> $\frac{5}{q} - \frac{5}{1-q} = 0$

=> $\frac{5}{q} = \frac{5}{1-q}$

=> $1 - q = q$

=> $2q = 1$

=> $\boldsymbol{q} = \frac{\mathbf{1}}{\mathbf{2}}$

Therefore, the Maximum Likelihood Estimation of $\boldsymbol{q} = \frac{\mathbf{1}}{\mathbf{2}}$

Also,

$$\frac{\partial^2}{\partial q^2} = \frac{-5}{q^2} - \frac{5}{(1-q)^2} < 0$$

Hence, at $q = \frac{1}{2}$, it will maximize the likelihood function of $X_1, \dots\ , X_n; \quad n = 10$

Solution 4:

$x = (x_1, \dots, x_N)^T$

$y = (y_1, \dots, y_N)^T$

From Bayes Theorem,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$ -------(1)

$$P(y|x,w,\beta) = \mathcal{N}\big(y|f(x,w),\beta^{-1}\big) = \prod_{n=1}^{N}\left[\frac{1}{\sqrt{2\pi\beta^{-1}}} * e^{\left[\frac{-1}{2}\left(\frac{f(x_n,w)-y_n}{\beta^{-1}}\right)^2\right]}\right]$$

Prior Gaussian Distribution

$$P(w|\alpha) = \mathcal{N}(w|0,\alpha^{-1}) = \left(\frac{\alpha}{2\pi}\right)^{\frac{(M+1)}{2}} * e^{\left(\frac{-\alpha}{2}w^T w\right)}$$

According to Bayes Theorem,

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

$$\propto \prod_{n=1}^{N}\left[\frac{1}{\sqrt{2\pi\beta^{-1}}} * e^{\left[\frac{-1}{2}\left(\frac{f(x_{n,w})-y_n}{\beta^{-1}}\right)^2\right]}\right] * \left(\frac{\alpha}{2\pi}\right)^{\frac{(M+1)}{2}} * e^{\left(\frac{-\alpha}{2}w^T w\right)}$$

$$\boldsymbol{P(w|x,y,\alpha,\beta) \propto \left[\frac{1}{\sqrt{2\pi\beta^{-1}}}\right]^N * e^{\left[\frac{-\beta}{2}\sum_{n=1}^{N}(f(x_{n,w})-y_n)^2\right]} * \left(\frac{\alpha}{2\pi}\right)^{\frac{(M+1)}{2}} * e^{\left(\frac{-\alpha}{2}w^T w\right)}} \quad \text{--(2)}$$

Taking negative logarithm throughout the equation (2),

$$-\log(P(w|x,y,\alpha,\beta)) \propto -\log\left[\left[\frac{1}{\sqrt{2\pi\beta^{-1}}}\right]^N * e^{\left[\frac{-\beta}{2}\sum_{n=1}^{N}(f(x_{n,w})-y_n)^2\right]} * \left(\frac{\alpha}{2\pi}\right)^{\frac{(M+1)}{2}} * e^{\left(\frac{-\alpha}{2}w^T w\right)}\right]$$

$$\propto N\log\left(\frac{2\pi}{\beta}\right)^{\frac{1}{2}} - \log\left[e^{\left[\frac{-\beta}{2}\sum_{n=1}^{N}(f(x_{n,w})-y_n)^2\right]}\right] + \left(\frac{(M+1)}{2}\right)\log\left[\frac{2\pi}{\alpha}\right] - \log\left[e^{\left(\frac{-\alpha}{2}w^T w\right)}\right]$$

$$\boldsymbol{-\log(P(w|x,y,\alpha,\beta)) \propto N\log\left(\frac{2\pi}{\beta}\right)^{\frac{1}{2}} + \frac{\beta}{2}\sum_{n=1}^{N}(f(x_{n,w})-y_n)^2 + \left(\frac{(M+1)}{2}\right)\log(2\pi) - \left(\frac{(M+1)}{2}\right)\log(\alpha) + \frac{\alpha}{2}w^T w} \quad \text{------(3)}$$

From equation (3) we can conclude that $\boldsymbol{N\log\left(\frac{2\pi}{\beta}\right)^{\frac{1}{2}}}$, $\boldsymbol{\left(\frac{(M+1)}{2}\right)\log(2\pi)}$, $\boldsymbol{\left(\frac{(M+1)}{2}\right)\log(\alpha)}$ are not depended on $\boldsymbol{w}$ after taking the negative logarithm throughout the equation (2). So, we don't consider those parts.

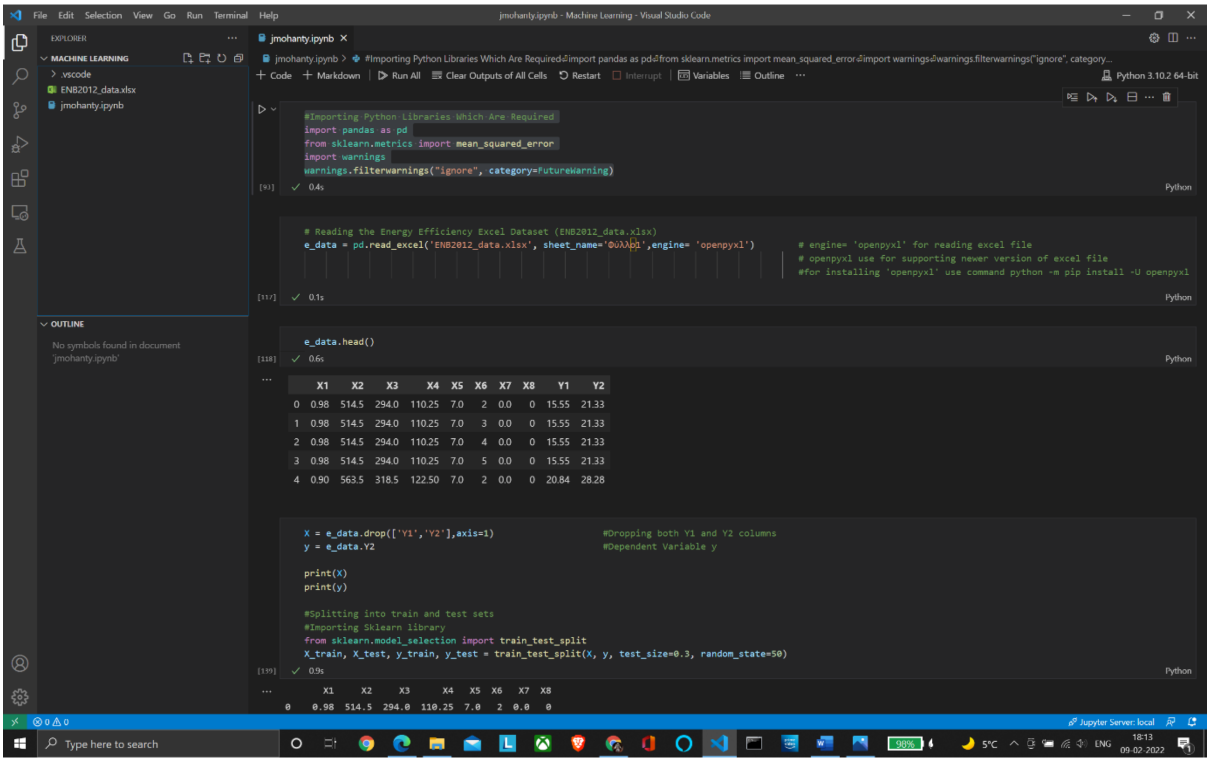
Therefore, we can now say that maximum posterior (MAP) is equivalent to minimizing the regularized sum-of-squares error function which is:

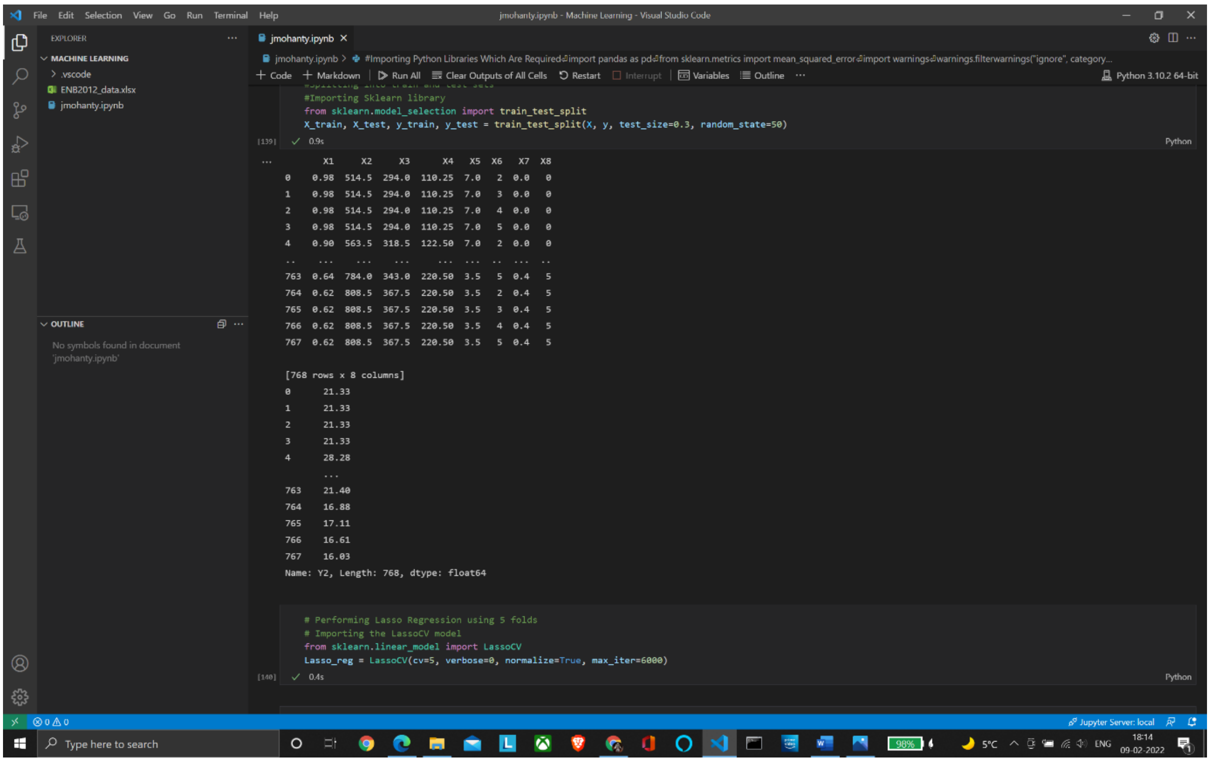
$$\boldsymbol{\frac{\beta}{2}\sum_{n=1}^{N}(f(x_{n,w})-y_n)^2 + \frac{\alpha}{2}w^T w}$$

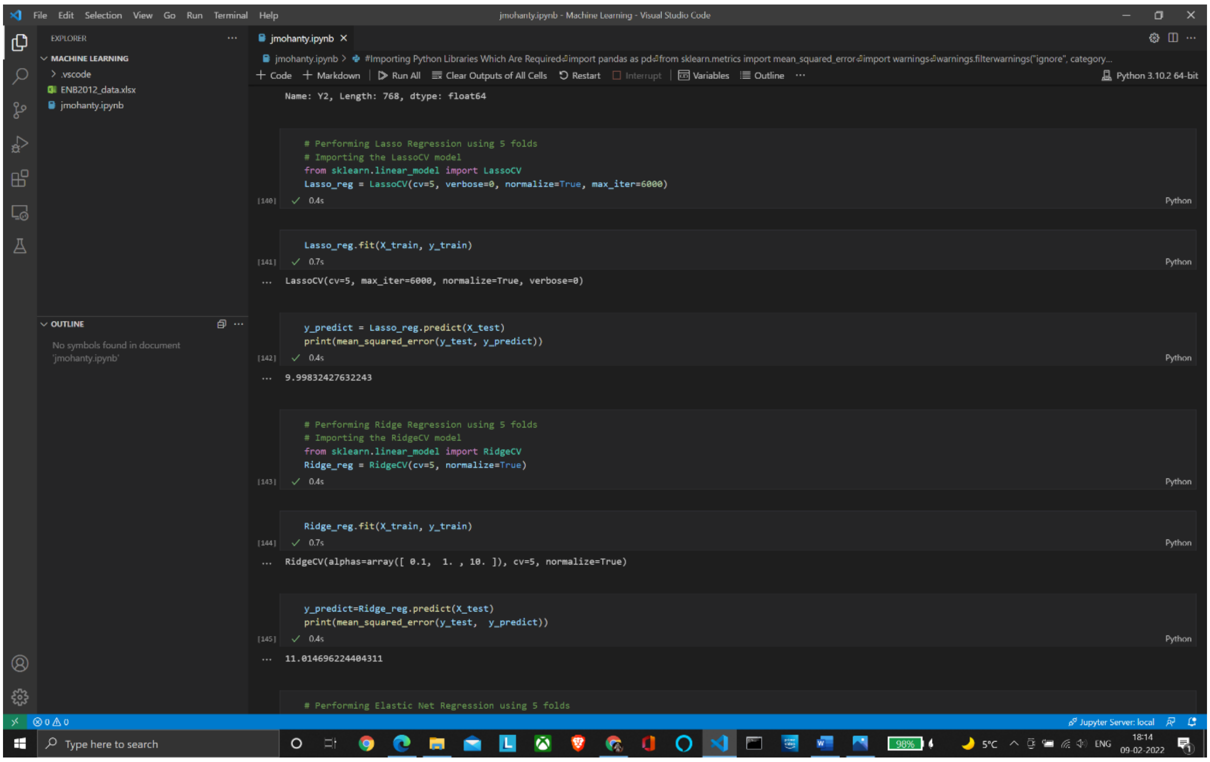
Solution 5:

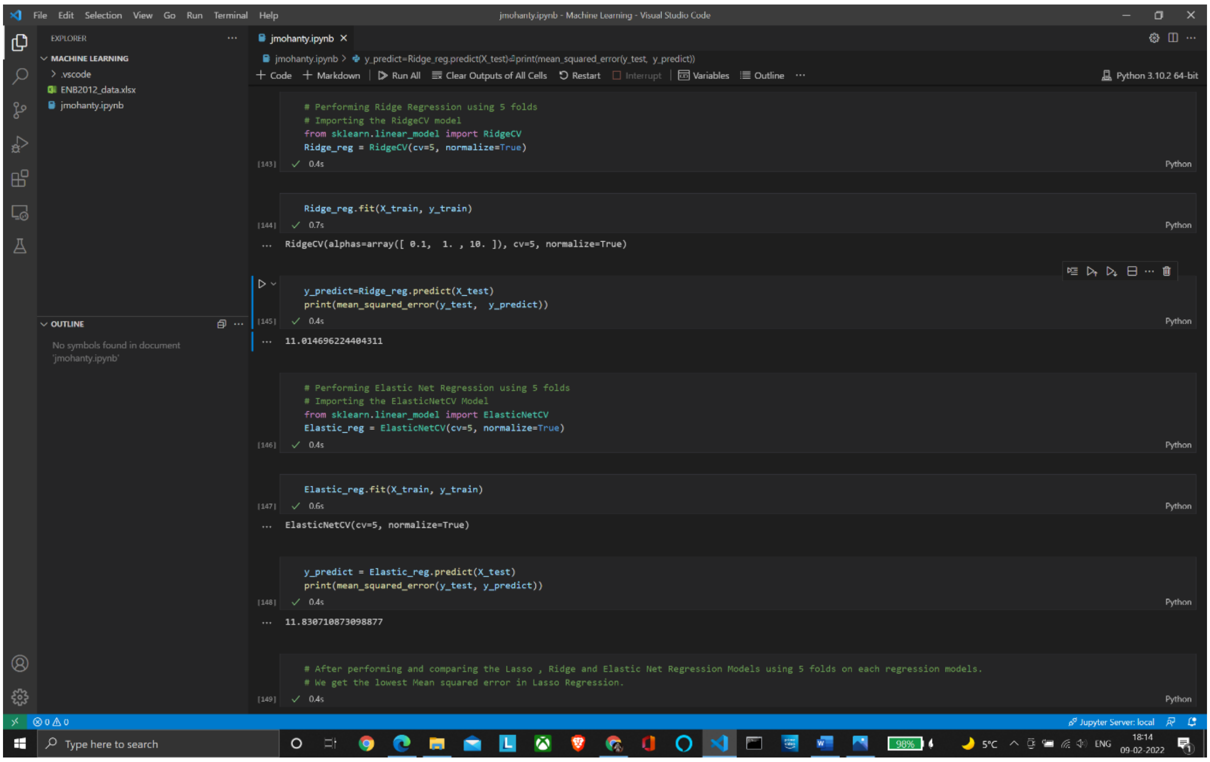
```
# After performing and comparing the Lasso , Ridge and Elastic Net Regression Models using 5 folds on each regression models.
# We get the lowest Mean squared error in Lasso Regression.
```

Screenshots




jmohanty.ipynb - Machine Learning - Visual Studio Code
#Importing Sklearn library
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=50)
[139] 0.9s
X1 X2 X3 X4 X5 X6 X7 X8
0 0.98 514.5 294.0 110.25 7.0 2 0.0 0
1 0.98 514.5 294.0 110.25 7.0 3 0.0 0
2 0.98 514.5 294.0 110.25 7.0 4 0.0 0
3 0.98 514.5 294.0 110.25 7.0 5 0.0 0
4 0.90 563.5 318.5 122.50 7.0 2 0.0 0
763 0.64 784.0 343.0 220.50 3.5 5 0.4 5
764 0.62 808.5 367.5 220.50 3.5 2 0.4 5
765 0.62 808.5 367.5 220.50 3.5 3 0.4 5
766 0.62 808.5 367.5 220.50 3.5 4 0.4 5
767 0.62 808.5 367.5 220.50 3.5 5 0.4 5
[768 rows x 8 columns]
0 21.33
1 21.33
2 21.33
3 21.33
4 28.28
763 21.40
764 16.88
765 17.11
766 16.61
767 16.03
Name: Y2, Length: 768, dtype: float64
# Performing Lasso Regression using 5 folds
# Importing the LassoCV model
from sklearn.linear_model import LassoCV
Lasso_reg = LassoCV(cv=5, verbose=0, normalize=True, max_iter=6000)
[140] 0.4s


jmohanty.ipynb - Machine Learning - Visual Studio Code
Name: Y2, Length: 768, dtype: float64
# Performing Lasso Regression using 5 folds
# Importing the LassoCV model
from sklearn.linear_model import LassoCV
Lasso_reg = LassoCV(cv=5, verbose=0, normalize=True, max_iter=6000)
[140] 0.4s
Lasso_reg.fit(X_train, y_train)
[141] 0.7s
LassoCV(cv=5, max_iter=6000, normalize=True, verbose=0)
y_predict = Lasso_reg.predict(X_test)
print(mean_squared_error(y_test, y_predict))
[142] 0.4s
9.99832427632243
# Performing Ridge Regression using 5 folds
# Importing the RidgeCV model
from sklearn.linear_model import RidgeCV
Ridge_reg = RidgeCV(cv=5, normalize=True)
[143] 0.4s
Ridge_reg.fit(X_train, y_train)
[144] 0.7s
RidgeCV(alphas=array([0.1, 1. , 10.]), cv=5, normalize=True)
y_predict=Ridge_reg.predict(X_test)
print(mean_squared_error(y_test, y_predict))
[145] 0.4s
11.014696224404311
Performing Elastic Net Regression using 5 folds


jmohanty.ipynb - Machine Learning - Visual Studio Code
# Performing Ridge Regression using 5 folds
# Importing the RidgeCV model
from sklearn.linear_model import RidgeCV
Ridge_reg = RidgeCV(cv=5, normalize=True)
Ridge_reg.fit(X_train, y_train)
RidgeCV(alphas=array([0.1, 1. , 10.]), cv=5, normalize=True)
y_predict=Ridge_reg.predict(X_test)
print(mean_squared_error(y_test, y_predict))
11.014696224404311
# Performing Elastic Net Regression using 5 folds
# Importing the ElasticNetCV Model
from sklearn.linear_model import ElasticNetCV
Elastic_reg = ElasticNetCV(cv=5, normalize=True)
Elastic_reg.fit(X_train, y_train)
ElasticNetCV(cv=5, normalize=True)
y_predict = Elastic_reg.predict(X_test)
print(mean_squared_error(y_test, y_predict))
11.830710873098877
After performing and comparing the Lasso , Ridge and Elastic Net Regression Models using 5 folds on each regression models.
We get the lowest Mean squared error in Lasso Regression.