Q3 - Linear Regression

Import Libraries

```
from sklearn.model_selection import LeaveOneOut, cross_val_predict
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import numpy as np
import math
import sys
import os
```

(a) Given the following three data points of (x, y): (1, 2), (2, 1), (0, -1), try to use a linear regression $y = \beta_0 + \beta_1 x$ to predict y. Determine the values of β_1 and β_0 and show each step of your work.

x	y	xy	x^2	y^2
1	2	2	1	4
2	1	2	4	1
0	-1	0	0	1
$\sum_i x_i = 3$	$\sum_i y_i = 2$	$\sum_i x_i y_i = 4$	$\sum_i x_i^2 = 5$	$\sum_i y_i^2 = 6$

We want to predict the dependent random variable y , and it is given by $y' = \beta_0 + \beta_1 x$

$\beta_0 = \frac{\sum_i y_i \sum_i x_i^2 - \sum_i x_i \sum_i x_i y_i}{n(\sum_i x_i^2) - (\sum_i x_i)^2} = \frac{2*5-3*4}{3*5-3*3} = \frac{-1}{3}$

$\beta_1 = \frac{n(\sum_i x_i y_i) - \sum_i x_i \sum_i y_i}{n(\sum_i x_i^2) - (\sum_i x_i)^2} = \frac{3*4-3*2}{3*5-3*3} = 1$

Thus the equation is $y' = \frac{-1}{3} + x$

(b) Linear Regression Programming Assignment

Apply the following three linear regressions: (1) $y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_4 x_4$ ***(2)*** $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$ ***(3)*** $y = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \gamma_3 x_3 + \gamma_4 x_4$ ***to the provided data file "hw3q3(b).csv", which is from a combined cycle power plant dataset (https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+Plant). In the given data file, xi are features and y is the prediction target which indicates hourly electrical energy output.***

(i) Load the data. Fit the whole dataset to the three linear regression models, respectively. Report the coefficients (alphas, betas, gammas) of the three models.

```
def data_and_headers(filename):
    data = None
    with open(filename) as fp:
        data = [x.strip().split(',') for x in fp.readlines()]
    headers = data[0]
    headers = np.asarray(headers)
    class_field = len(headers) - 1
    data_x = [[float(x[i]) for i in range(class_field)] for x in data[1:]]
    data_x = np.asarray(data_x)
    data_y = [[float(x[i]) for i in range(class_field, class_field + 1)] for x in data[1:]]
    data_y = np.asarray(data_y)
    return headers, data_x, data_y
```

```
headers, features_x, labels_y = data_and_headers('Data' + os.sep + 'hw3q3(b).csv')
```

```
modela = LinearRegression().fit(features_x, labels_y.flatten())
modelb = LinearRegression().fit(features_x**2, labels_y.flatten())
modelc = LinearRegression().fit(features_x**3, labels_y.flatten())
print('Coefficients of Simple LR - \t{}, Intercept - {:.4f}'.format(modela.coef_,modela.intercept_))
print('Coefficients of Quadratic LR - \t{}, Intercept - {:.4f}'.format(modelb.coef_,modelb.intercept_))
print('Coefficients of Cubic LR - \t{}, Intercept - {:.4f}'.format(modelc.coef_,modelc.intercept_))
```

Output -

```
Coefficients of Simple LR -     [-12.38926535   2.80059786 -12.32760055
-64.67916351], Intercept - 500.2071
Coefficients of Quadratic LR -  [ -6.05581029   7.28426322 -15.38358205
-54.34236701], Intercept - 477.0979
Coefficients of Cubic LR -  [  1.24877688  16.3800381  -24.20328807
-43.63328247], Intercept - 466.2837
```

The above output can be interpreted as follows, where $c_i \in \{\alpha_i, \beta_i, \gamma_i\}$ and $i \in \{0, 1, 2, 3, 4\}$-

Equation	c_0	c_1	c_2	c_3	c_4
$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_4 x_4$	500.2071	-12.3892	2.8005	-12.3276	-64.6791
$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$	477.0979	-6.0558	7.2842	-15.3836	-54.3423
$y = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \gamma_3 x_3 + \gamma_4 x_4$	466.2837	1.2487	16.3800	-24.2032	-43.6333

(ii) Use leave-one-out cross validation to determine the RMSE (root mean square error) for the three models. Specifically, in each fold, fit the training data to the model to determine the coefficients, then apply the coefficients to get predicted label for testing data (You don't need to report the coefficients in each fold). Report RMSE for the three models. Based on the RMSE, which model is the best for fitting the given data?

```
model1 = LinearRegression()
model2 = LinearRegression()
model3 = LinearRegression()
loocv = LeaveOneOut()
ypred1 = cross_val_predict(model1, features_x, labels_y.flatten(), cv=loocv)
ypred2 = cross_val_predict(model2, features_x**2, labels_y.flatten(), cv=loocv)
ypred3 = cross_val_predict(model3, features_x**3, labels_y.flatten(), cv=loocv)
print('Normal LR RMSE - {:.4f}'.format(math.sqrt(mean_squared_error(labels_y.flatten(), ypred1))))
print('Quadartic LR RMSE - {:.4f}'.format(math.sqrt(mean_squared_error(labels_y.flatten(), ypred2))))
print('Cubic LR RMSE - {:.4f}'.format(math.sqrt(mean_squared_error(labels_y.flatten(), ypred3))))
```

Output -

```
Normal LR RMSE - 4.4927
Quadartic LR RMSE - 6.4587
Cubic LR RMSE - 8.0864
```

Based on the RMSE, simple linear regression is the best for fitting the given data.