REPUBLIC OF CAMEROON

Peace – Work – Fatherland

THE UNIVERSITY OF BAMENDA

P.O. Box 39, Bambili

Tel: (237) 233 366 033 / 233 366 029

Fax: (237) 233 366 030

Website: www.unibda.edu.cm,

Email: info@unibda.edu.cm

REPUBLIQUE DU CAMEROUN

Paix – Travail – Patrie

L'UNIVERSITÉ DE BAMENDA

B.P. 39, Bambili

Tel: (237) 233 366 033 / 233 366 029

Fax: (237) 233 366 030

Website: www.unibda.edu.cm,

Email: info@unibda.edu.cm

COURSE TITLE: MACHINE LEARNING

GROUP NUMBER: L4 Catch Up

LEVEL: 500

COURSE INSTRUCTOR: Eng. Kunde Godfrey

Machine Learing Group L4 Catchup

GROUP MEMBERS

Names	Matricule
Soh Juvitus Leong	UBa19E0205
Fonsi Royce Apuagho	UBa19E0170
Kiyong Desmond Yuochi Junior	Uba17E0001
Chi Thierry	UBa19E0008
Nji Ruth lum	UBa18E0027

QUESTION:

"Instructions: Your are required to complete this assessment before our next class. You shall also present your result in the next class. The document is 15 mks and the presentation 10 mks.

Question

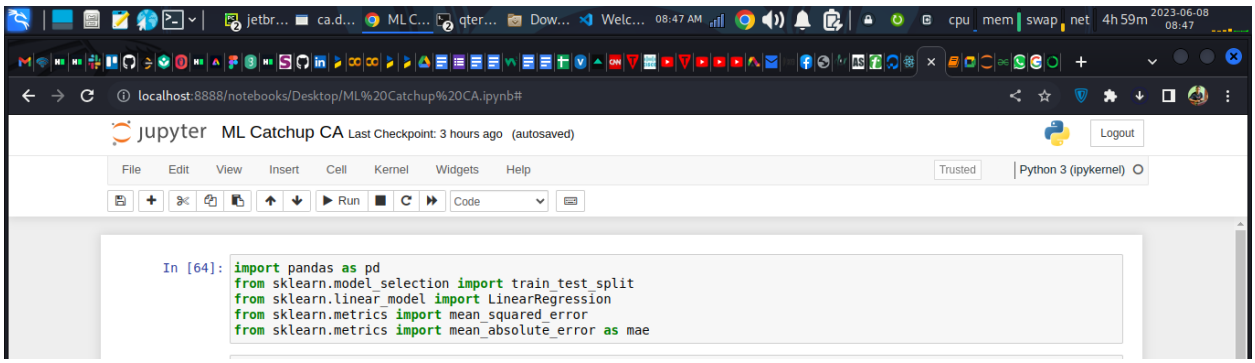
The cleaned dataset 'concrete_clean.csv' having a total of 968 data and 10 columns (cement, water, coarse aggregate, fine aggregate, age of testing, fly ash, superplasticizer, blast furnace slag, w/c ration & strength)
Design implement and the evaluate linear regression model to predict the comparative strength of concrete using this dataset. The data is already cleaned.
Use the mean square error and mean absolute error to evaluate the performance of your solution"

Language Used: Python
Framework: Jupyter Notebook

Brief Description of Procedure
Import the necessary libraries
Read CSV file into DataFrame
Split data into training and testing models
Train the model
Do some predictions
Run a Mean square error and mean absolute error to evaluate model

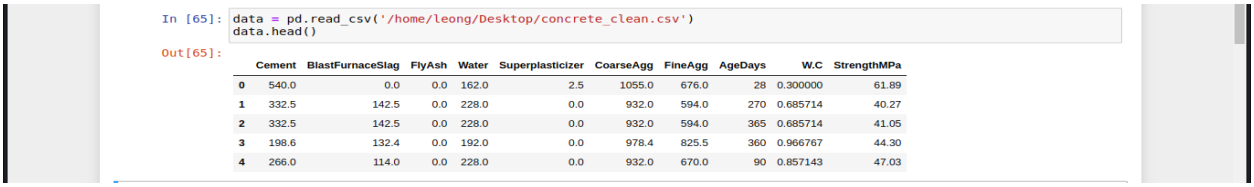
Detail procedure

1) Import Libraries:



import pandas as pd: The pandas library for processing data
from sklearn.model_selection import train_test_split: to train the model
from sklearn.linear_model import LinearRegression: THe algorithm used for this is the Linear Regression algorithm
from sklearn.metrics import mean_squared_error: Library to perform the mean square error evaluation
from sklearn.metrics import mean_absolute_error as mae: library to perform the mean absolute error evaluation

2) **Reading the CSV file**



	Cement	BlastFurnaceSlag	FlyAsh	Water	Superplasticizer	CoarseAgg	FineAgg	AgeDays	W.C	StrengthMPa
0	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	0.300000	61.89
1	332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	0.685714	40.27
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	0.685714	41.05
3	198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	0.966767	44.30
4	266.0	114.0	0.0	228.0	0.0	932.0	670.0	90	0.857143	47.03

Read the file from csv in the directory which it is stored in the laptop locally

3) **Split data into training and Testing model:**

```
4   266.0   114.0   0.0   228.0   0.0   932.0   670.0   90   0.857143   47.03

In [88]: X = data.drop(columns=['StrengthMPa'])
         Y = data['StrengthMPa']

In [89]: #random_state controls the shuffling of the set selected at random for the split
         X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state=2)
```

From the data, we are to test the strength of concrete. Hence we use the concrete strength column for our testing sample and then the rest of the tables used as training data sets

4) Train the model:

```
In [89]: #random_state controls the shuffling of the set selected at random for the split
         X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state=2)

In [90]: lin_reg = LinearRegression()
         lin_reg.fit(X_train, Y_train)
```

We define the X_train, X_test, Y_train, Y_test data sets using the train_test_split model and assign a test size of 20% of the whole data. A random state variable of 2 is assigned to it. This helps to control the shuffling process when splitting the data into varios samples for training and testing

```
In [69]: lin_reg = LinearRegression()
         lin_reg.fit(X_train, Y_train)

Out[69]: ▾ LinearRegression
         LinearRegression()
```

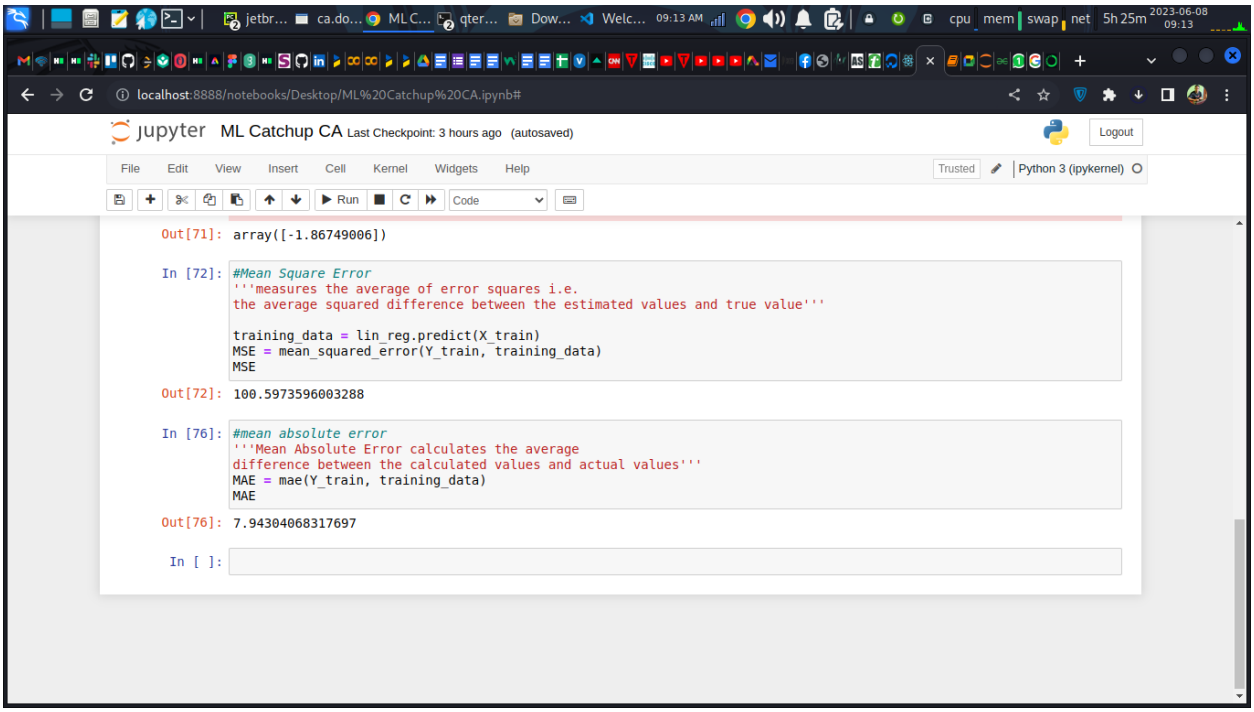
Here we call the linear regression model and fit the data to train into it and it is trained.

5) Make predictions:

```
In [70]: #predicting
         predictions1 = lin_reg.predict([[445.8, 34.8, 0.1, 123.0, 1.2, 456.9, 213.4, 98, 0.606769]])
         predictions1

         /home/leong/.local/lib/python3.11/site-packages/sklearn/base.py:439: UserWarning: X does not have valid feature na
         mes, but LinearRegression was fitted with feature names
           warnings.warn(

Out[70]: array([42.7772228])

In [71]: predictions2 = lin_reg.predict([[115.8, 24.8, 0.0, 123.0, 1.9, 206.9, 111.4, 15, 0.903369]])
         predictions2

         /home/leong/.local/lib/python3.11/site-packages/sklearn/base.py:439: UserWarning: X does not have valid feature na
         mes, but LinearRegression was fitted with feature names
           warnings.warn(

Out[71]: array([-1.86749006])
```

Run some predictions and get the results

6) Evaluations:



a) Mean square error: This measures the average of error squares i.e the average squared difference between the estimated values and true value. And for this machine model, it gave a value of 97.0731726465731. Meaning the squared mean of the distance separating the actual and predicted values is 97.0731726465731

b) Mean absolute error: calculates the average difference between the predicted values and actual values. This figure came out to be 7.766144523341548