School of Computer Science Engineering and Technology

Course- BCA
Course Code- BCA356L
Year-2020
Date- 16-02-2022

Type- Elective
Course Name- Applications of Data Science
Semester- Even
Batch- NA

Lab Assignment/4.1

Objective: Understanding the concept of Linear Regression

Python Packages used in this Assignment

- **import numpy as np**

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Link: https://numpy.org/

- **import matplotlib.pyplot as plt**

matplotlib.pyplot is a state-based interface to matplotlib. It provides an implicit, MATLAB-like, way of plotting. It also opens figures on your screen, and acts as the figure GUI manager.

pyplot is mainly intended for interactive plots and simple cases of programmatic plot generation:

Link: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.html

- **from sklearn.linear_model import LinearRegression**

Scikit-learn is an open source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection, model evaluation, and many other utilities.

LinearRegression fits a linear model with coefficients w = (w1, …, wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

Link: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

Methods

fit(X, y[, sample_weight])	Fit linear model.
get_params([deep])	Get parameters for this estimator.
predict(X)	Predict using the linear model.
score(X, y[, sample_weight])	Return the coefficient of determination of the prediction.
set_params(**params)	Set the parameters of this estimator.

- **from sklearn import metrics**

Metric functions: The sklearn.metrics module implements functions assessing prediction error for specific purposes. These metrics are detailed in sections on Classification metrics, Multilabel ranking metrics, Regression metrics and Clustering metrics.

Link: https://scikit-learn.org/stable/modules/model_evaluation.html

- **from sklearn.model_selection import train_test_split**

Split arrays or matrices into random train and test subsets.

Link: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

- **from sklearn import datasets**

The sklearn.datasets module includes utilities to load datasets, including methods to load and fetch popular reference datasets. It also features some artificial data generators.

Link: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets

Problems

Problem 1: Linear Regression (own Python code) (40 minutes)

A patient is given a drip feed containing a particular chemical and its concentration in his blood is measured, in suitable units, at one hour intervals. The doctors believe that a linear relationship will exist between the variables.

Time, x (hours)	0	1	2	3	4	5	6
Concentration, y	2.4	4.3	5.0	6.9	9.1	11.4	13.5

Concentration is the dependent variable as the concentration in the blood is likely to vary according to time.

The doctor may wish to estimate the concentration of the chemical in the blood after 3.5 hours.

a) Write the Python code (do not use Linear Regression of the sklearn package) to demonstrate the working of Linear Regression and obtain the blood concentration after 3.5 hours

b) Also, compare your regression model with the model obtain while using Linear Regression of the sklearn package.

Problem 2: Linear Regression (20 minutes)

The heights and weights of a sample of 11 students are:

Height (m) h	1.36	1.47	1.54	1.56	1.59	1.63	1.66	1.67	1.69	1.74	1.81
Weight (kg) w	52	50	67	62	69	74	59	87	77	73	67

Use the above (in Problem 1) created regression model to obtain

a) The regression line of w on h.

b) Use the regression line to estimate the weight of someone whose height is 1.6m.

Problem 3: Multiple Linear Regression (30 minutes)

Use 'boston' dataset using 'from sklearn import datasets', as below

boston = datasets.load_boston(*, return_X_y=False)

It, load and return the boston house-prices dataset (regression).

Samples total	506
Dimensionality	13
Features	real, positive

We can see, the dimensionality of this data is 13. Now, perform

a) Use this data and split it into training and testing data

b) Use linear regression to fit this data.

c) After that, obtain "regression coefficients".

d) plot residual errors in training data

e) plot residual errors in testing data

School of Computer Science Engineering and Technology

Submission Instructions:

- Submission required a zip file containing
  a) Python file (Roll_No_Lab3.ipynb/.py)
  b) PDF of .ipynb file (only if submitting ipynb file).
- Use the naming convention: RollNo_Lab4 (Example: 123_Lab3))
- Submission is through LMS only.