CENG313 Introduction to Data Science
Fall 2022-2023
Lecturer: Dr. Duygu Sarıkaya
Teaching Assistant: Berrin İşlek
Gazi University, Department of Computer Engineering
Assignment 2 is due the 5th of December 2022, Monday 23:59
Assignment 2: Regression - Iris Flowers

Linear Regression analysis is used to predict the value of a variable based on the value of another variable [REF: https://www.ibm.com/topics/linear-regression]. It is a type of regression model and a supervised learning algorithm in Machine Learning. You'll find that linear regression is used in everything from biological, behavioral, environmental, and social sciences to business. Linear-regression models have become a proven way to scientifically and reliably predict the future. Because linear regression is a long-established statistical procedure, the properties of linear-regression models are well understood and can be trained very quickly.

In this assignment, you will be working on Linear Regression to predict the petal length (cm) of Iris flowers. We will use the Iris Flower Dataset with the features: Petal Width and Petal Length. For this assignment, we will use all 150 flowers.

Iris Dataset:

The Iris flower data set or Fisher's Iris data set is a multivariate data set. The data is collected to quantify the morphologic variation of Iris flowers of three related species. The dataset consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor, where total number of samples is 150). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters. You can load the Iris dataset using scikit-learn (sklearn.datasets. load_iris)

For this assignment, you will need these libraries: pandas, NumPy, scikit-learn (sklearn), matplotlib and seaborn.

Predicting the Petal Length:

1. Plot (scatter plot matrix) the Iris flower dataset. The matrix should include individual scatter plots for every combination of variables in the Iris flower dataset. A scatter plot matrix can help you observe linear correlations/relationships between multiple variables. In the diagonal, you should show the histograms of each variable. You should also color code each data instance (Iris flower) on the scatter plots. You can use seaborn to visualize.

After this point, you will use only two features: Petal Width and Petal Length. Please remove any other columns from the dataset.

2. You will **split the dataset** into two sets: train and test set. You should first shuffle your data, and then split %70 of the dataset for training and %30 for testing.
3. You will train a Linear Regression model that learns from the training set **to predict the Petal Length** (cm) of the Iris flowers. You can use scikit-learn for this.
4. You will test your model on the test set. For all the instances (=flowers) in the test set, you will be computing and outputting two different metrics: the Mean Absolute Error and the Mean Squared Error. In order to compute these error values, you should **compare the predicted value** for the petal length and **the actual value** for the petal length.

5. You will pick a random flower from the test set, and output its features. Then, you should use the trained model **to predict the petal length of this specific flower**. Please output the predicted value for the petal length and the actual value for the petal length. Calculate and output the difference between the predicted and actual values. Then calculate and output the Mean Squared Error for this specific test.

Fitting Polynomials and R^2 Analysis:

6. You will fit a polynomial of degree 1, basically a line, to the data using the polyfit() function from NumPy, and output the equation in the form of mx + b = 0.
7. Plot the polynomial of degree (line) on a scatter plot that shows the data instances (Iris flowers) and their Petal Width and Petal Length values on the axes.
8. Model the data using the parameters of the fitted straight line, and then compute and output the value of R^2, which measures how well the model (the straight line) fits the data.
9. You will now fit a second-degree polynomial, basically a curve, to the data using the polyfit() function from NumPy, and output the equation in the form of ax2 + bx+ c = 0.
10. Plot the second-degree polynomial (curve), on a scatter plot that shows the data instances (Iris flowers) and their Petal Width and Petal Length values on the axes.

Important Note:

You will receive points only if your script executes, shows the correct answer, and includes the explanation (text in the comment section at the top of each section). **This is an individual assignment**, meaning that you will be working on it alone (please check the **Class Rules and Expectations** below, also available in the syllabus)

Submission:

You will submit a jupyter notebook (ipynb file) with an executable Python script with comments that explain the code. You can zip your file (or rar) when you submit on guzem as it may not allow ipynb extensions.

You should import all the libraries you will use at the top of your notebook. Please refer to course slides, tutorials, and practicals to set up a running Python environment, Jupyter notebook and to import these libraries. You can check the documentation of each library (available online) to get more information about the functions you will use. Important Note: Please submit your file name in this format: Studentno_StudentName_StudentSurname

Grading:

The total is 100 points. **You will receive points only if** your script executes, shows the correct answer, and includes the explanation (text in the comment section at the top of each section)

Course Rules and Expectations

All work on programming assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, however, everything that is turned in for each assignment must be your own work. In particular, it is not acceptable to: submit another person’s assignment as your own work (in part or in its entirety), get someone else to do all or a part of the work for you, submit a previous work that was done for another course in its entirety (self- plagiarism), submit material found on the web as is etc. **Important Note: Material found online and used as is will lead to your code being similar to many others.** These acts are in violation of academic integrity (plagiarism), and these incidents will not be tolerated. Homeworks, programming assignments, exams and projects are subject to Turnitin and Moss (Measure of Software Similarity) checks. Use sources to learn from only, and write your own code from scratch.