DHRUV GARG 16 BCE 1190

LOGISTIC REGRESSION USING SKLEARN

LAB 4

Logistic regression theory:

Logistic regression is used to predict the outcome of a **categorical variable**. A categorical variable is a variable that can take only specific and limited values. Logistic regression fits the data points using the sigmoid function. The **formula of the sigmoid function** is given as follows:

$$S(x) = \frac{1}{1+e^{-x}}$$

The Sigmoid function has an S-shaped curve. It has a finite limit of 0 as x approaches negative infinity and 1 as x approaches positive infinity.

While using the sklearn's logistic regression class, we can obtain better fitting to our data by making changes to **parameters** such as "solver", "c" and "random_state".

PART 1: SEEDS Dataset

First, we visualized the sigmoid function that is used by the logistic regression model. This gives an S-shaped curve between 0 and 1.

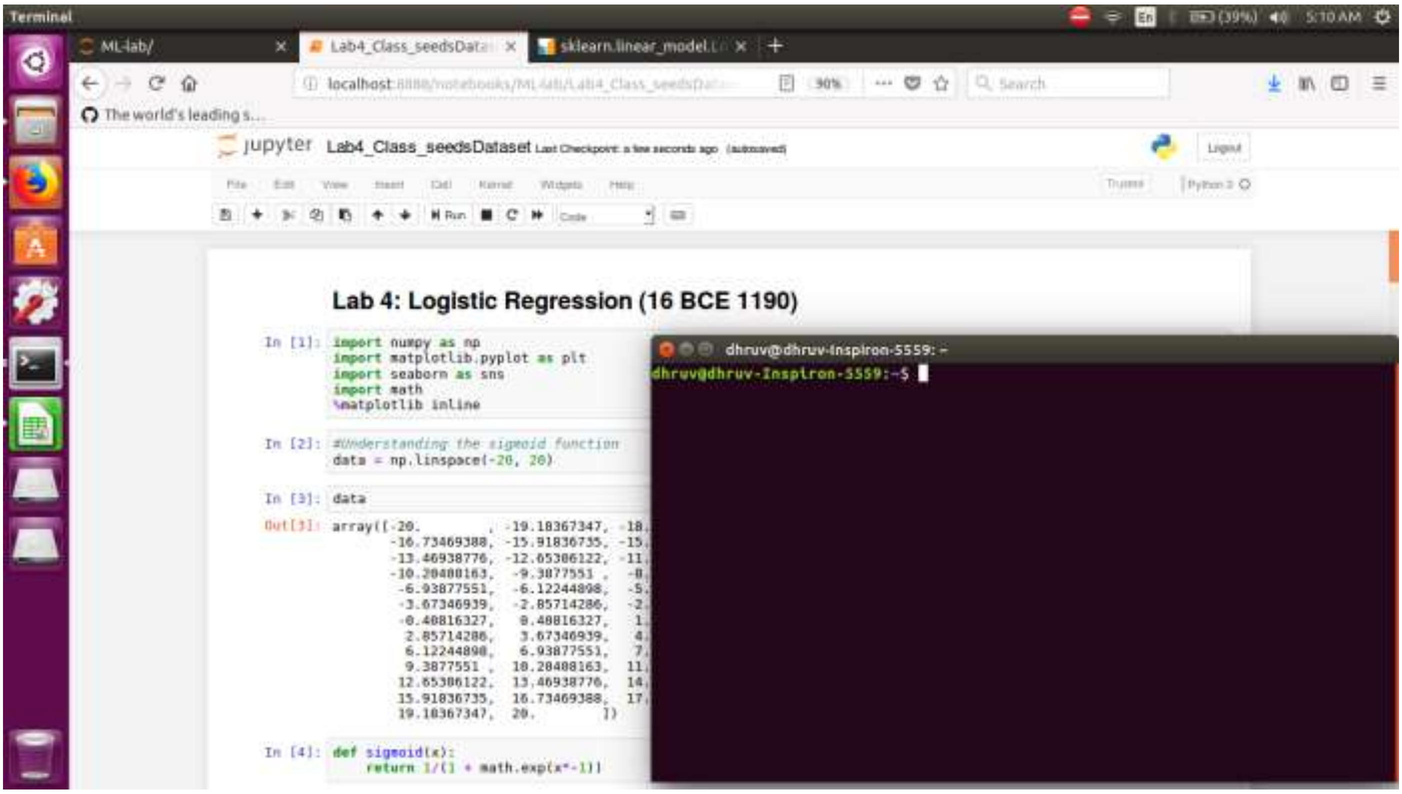
Next, we work on the seeds dataset. We do basic **pre-processing** of the dataset such as checking for null values and scaling the values. Since we need to do a classification, we must identify the parameters to use. To do this, we plot pairplots using seaborn. We observe that the 3 classes of seeds can be classified using the length_of_the_groove and length_of_kernel attributes. Next, using train_test_split, we partition the data into train set and test set. **First** we fit the train data to the logistic regression model **using default parameters**. At first an accuracy of 96.22% was obtained. To **increase the accuracy**, we **change the parameters**. I gradually increased the value of "c" and observed the increase in accuracy. Higher the "c" value, better tighter fitting of the data occurs. If we have a very large value of c, overfitting of data could occur. In my case, the **accuracy increased to 98.11%** after setting c as 2. After c = 2, it did not increase further.

PART 2: ABSENTEEISM FROM WORK Dataset

The dataset was already pre-processed and scaled, as a result of previous labs' work. In this exercise I **first** applied logistic regression to try to **classify 2 classes**. I had already extracted Graduate vs Doctorate data (at the time of classification using perceptron). We try to achieve similar classification using logistic regression. We apply the **"liblinear" model of the LogisticRegression** class to do this. Using default parameters, I got an accuracy of 92.3%. Further, on increasing "c" to get a higher accuracy, I got an accuracy of 100% at c = 5, using the same training-testing data and the same algorithm. **Next**, I performed **multi-class logistic regression** on categories of pets. The rows were extracted and the attributes that linearly separated the pet classes were identified. In this, I wanted to perform classification between 3 classes. **Since liblinear is not a good algorithm for multi-class, I used newton-cg and lbfgs** (Saga was also tried, but it was unable to converge). For both of these algorithms, the random_state was different, so as to assess the impact of different training datasets. For each of the two, once the **default parameters** were used[newton-cg accuracy: 71.4%, lbfgs: 92.85%] and then the parameter **"c" was tweaked** to improve the accuracy of the model. In both cases, the **accuracy improved** on increasing the value of c[newton-cg: 92.85% and lbfgs:100%].

SCREENSHOTS

Seeds-dataset

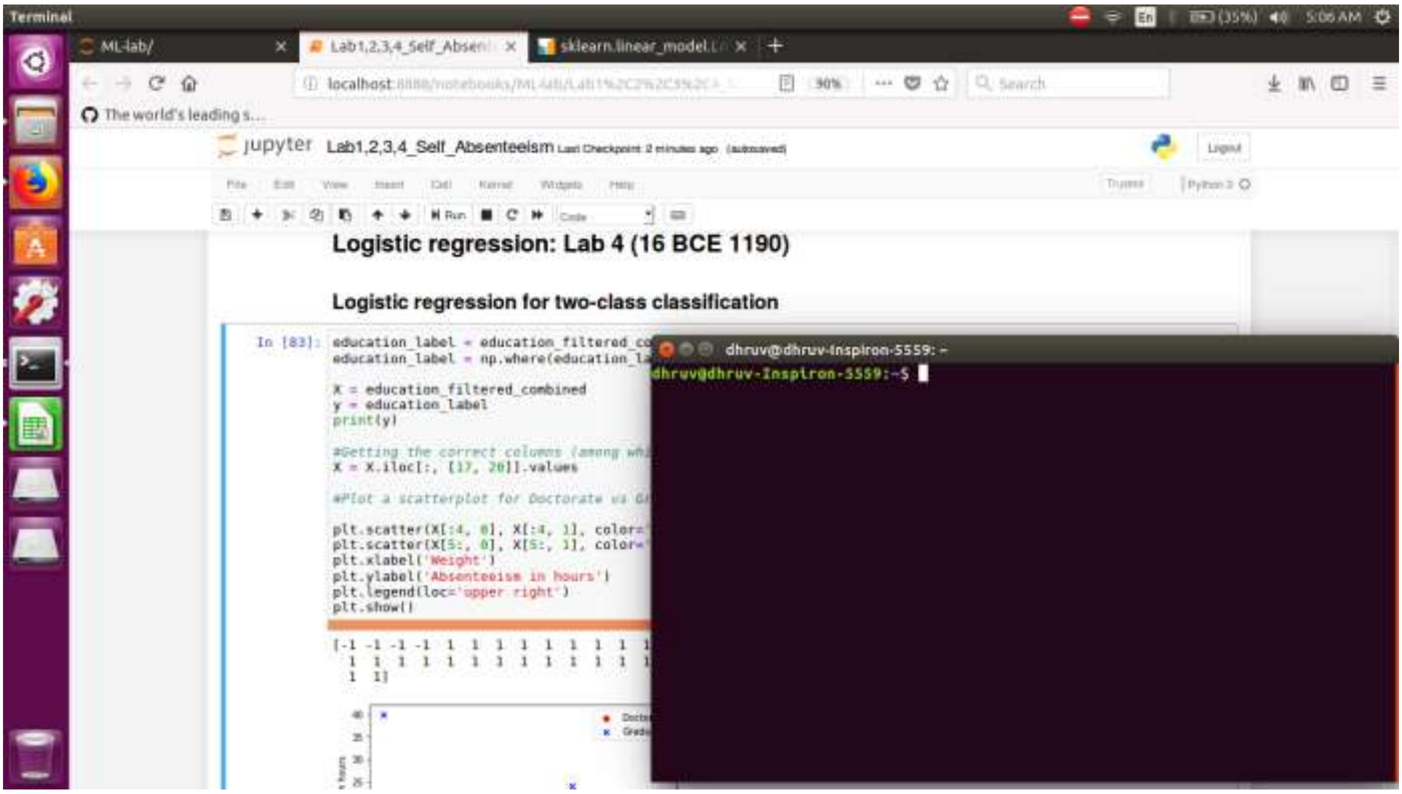


Lab 4: Logistic Regression (16 BCE 1190)

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        import math
        %matplotlib inline

In [2]: #Understanding the sigmoid function
        data = np.linspace(-20, 20)

In [3]: data

Out[3]: array([-20.        , -19.18367347, -18.
               -16.73469388, -15.91836735, -15.
               -13.46938776, -12.65306122, -11.
               -10.20408163,  -9.3877551 ,  -8.
                -6.93877551,  -6.12244898,  -5.
                -3.67346939,  -2.85714286,  -2.
                -0.40816327,   0.40816327,   1.
                 2.85714286,   3.67346939,   4.
                 6.12244898,   6.93877551,   7.
                 9.3877551 ,  10.20408163,  11.
                12.65306122,  13.46938776,  14.
                15.91836735,  16.73469388,  17.
                19.18367347,  20.        ])

In [4]: def sigmoid(x):
            return 1/(1 + math.exp(x*-1))
```

```
dhruv@dhruv-Inspiron-5559:~$
```

Chosen dataset – Absenteeism at work



Logistic regression: Lab 4 (16 BCE 1190)

Logistic regression for two-class classification

```
In [83]: education_label = education_filtered_co
         education_label = np.where(education_la

         X = education_filtered_combined
         y = education_label
         print(y)

         #Getting the correct columns (among wh
         X = X.iloc[:, [17, 20]].values

         #Plot a scatterplot for Doctorate vs Gr

         plt.scatter(X[:4, 0], X[:4, 1], color=
         plt.scatter(X[5:, 0], X[5:, 1], color=
         plt.xlabel('Weight')
         plt.ylabel('Absenteeism in hours')
         plt.legend(loc='upper right')
         plt.show()

[-1 -1 -1 -1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1]
```

```
dhruv@dhruv-Inspiron-5559:~$
```