REPORT ON LOGISTIC REGRESSION USING GRADIENT DESCENT

TASK 1

- The first function sigmoid() is generally called as the sigmoid function or the logistic function. This function generally returns value between 0 and 1. There are many different functions but sigmoid is very common and generally used especially in Logistic Regression. Logistic regression used the sigmoid to predict the output.
- The Second function is called as the cost function or the lost function. In my code it is defined as loss_func(). The cost function gives how far the predicted output is from the original output.
- We need to update the theta value (lambda values) for which gradient descent is used. My function grad_desc does just that.
- Finally, the pred function is used to calculate the accuracy and gives the prediction value.

TASK 2

- For this task I have used pandas to visualize and load the dataset.
- After I have loaded the dataset, I have created a variable which is our output variable which can take values depending on the condition as given in the question.
- Then, I created a copy of the data set so that I can create dummy variables for the origin variable as it is a qualitative variable. I used the in-built get_dummies function and the resulting dataset is as shown in the notebook.
- The next task was to normalize the attributes, for this I used the preprocessing library from sklearn which has a normalize function.
- I have used the drop function from pandas to drop the columns which were not required in the dataset and loaded X with attributes which were required.

TASK 3

- Task 3 was to just split the data into training and test set.
- I have used the train_test_split function for that to randomly split the data into 50-50% each. The random state used is the birth date as asked in the question

TASK 4

- In task 4, we are supposed to use different values for initial weights, learning rates and epochs.
- My Observation are as follows: -
    - For initial weight = 0.3, learning rate = 0.001 and epochs = 100, I am getting an error rate of 0.4 on the training set.
    - For initial weight = -0.5, learning rate = 0.00001 and epochs = 200, I am getting an error rate of 0.54 on the training set.
    - For initial weight = 0.6, learning rate = 0.01 and epochs = 15000, I am getting an error rate of 0.54 on the test set.
- The learning rate the step size, initial weight is lambda(theta) in my case and epochs is number of times the gradient descent has to run.