Text in red color is python code

Use naïve bayes classifier assignment data

1. Load data in numpy arrays

   For Training Data

   Train_X = np.loadtxt('trainX.txt', dtype=np.uint8)

   Create and Append col of 1's to training data

   Col_Of_Ones = np.ones((Train_X.shape[0]), dtype=np.uint8)

   Train_X = np.insert(Train_X , 0 , Col_Of_Ones , axis = 1)

   Map 2 and 4 classes to 1 and 0

   Train_Y = np.loadtxt('trainY.txt', dtype=np.uint8)

   Train_Y = np.where(Train_Y == 2 , 1 , 0)

   For Testing Data

   Do it yourself similarly as above

2. Randomly initialize weights between 0 and 1

   Thetas = np.random.uniform(0, 1, size=(-----think which given option is correct-------))

   { Train_X.shape[1] , Train_X.shape[0] , Train_Y.shape[1] , Train_Y.shape[0]}

3. How to take probabilities and prediction

   Probabilities = sigmoid(np.dot(Train_X , Thetas)) Sigmoid is define below

   Predictions = np.where(Probabilities >= 0.5 , 1 , 0)

4. How to calculate Accuracy

   calculate_Accuracy(Predictions , Train_Y) calculate_Accuracy is define below

5. Use loop and define stopping criteria of loop e.g loop is true until required accuracy is not achieved or for 1000 iteration

   while (not(Accuracy == 100) and (iterations <= 1000)):

   Probabilities = Do it yourself

   Predictions = Do it yourself

   update thetas / weights in loop by vectorized implementation

   Thetas = Thetas - (Eta / m) * np.dot(Train_X.T , (Probabilities - Train_Y))

   Calculate accuracy in loop

   Accuracy = calculate_Accuracy(Predictions , Train_Y)

6. Accuracy on Training and Testing

After loop calculate accuracy on training data and testing data and print them

Sigmoid function

```
def sigmoid(x):
return 1 / (1 + np.exp(-x))
```

Accuracy function

```
def calculate_Accuracy(Predictions , Actuals):
 return np.where(Predictions == Actuals , 1 , 0).sum() / Actuals.shape[0] * 100
```