PROJECT 1: Density Estimation and Classification

NAIVE BAYES CLASSIFIER:

Step 1:
Divide the dataset into training and testing data for the digit “7” and digit “ 8” respectively.

Step 2:
Estimation of Parameters: Extract features(mean and standard deviation) for every image in the training dataset. Mean and standard deviation become the new features for every image in the dataset. This leaves us with only two features instead of 784 features.

Step 3:
Estimation of values: Extract features(mean and standard deviation) for each image in the test dataset. Mean and standard deviation become the new features for every image in the dataset. This leaves us with only two features instead of 784 features.

Step 4:
Estimation of Normal Distribution Expressions: Calculate the mean and standard deviation for each of these new features which become the two parameters for 2-D normal distribution.

Step 5:
An explanation for distributions used in classification: Iterate through every image in the divided test dataset (differently for the digit “7” and digit “8”) to calculate the PDF(probability density function) of the digit “7” and digit “8” respectively. Following steps to be followed to find the required PDF of the two datasets respectively:

1. **Finding P(Y|X) :**

   Probability of a class given the dataset is given by Bayes Rule:

$$\underbrace{P(c\mid x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x\mid c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

The posterior probability has to be calculated for both the digits “7” and “8”.

2. Naive Bayes Classifier has conditional independence, that is, given the class labels, features are independent. Thus, the likelihood can be calculated as:

$$P(\mathbf{X}|Y) = P(x_1|Y)P(x_2|Y), \ldots, P(x_d|Y)$$

3. Class Prior can be calculated as follows:

$$\hat{P}(y) = \frac{|\{i : y_i = y\}|}{n}$$

4. Naive Bayes Decision Rule:
   To simplify the calculations, instead of calculating the formula in step 3.1, we calculate the following:

$$P(c\,|\,x) \quad = \log p(C_k) + \sum_{i=1}^{n} x_i \cdot \log p_{ki}$$

5. Calculate the mean and standard deviation for each of these new features which become the two parameters for 2-D normal distribution.
6. Insert the acquired values from all the above steps into the PDF of 2-D Normal Distribution for Bivariate Features. The Probability Density Function is calculated by the following formula:

$$\mathcal{N}\left(x|\mu, \sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

where μ is mean and σ is the standard deviation of the estimated features.

Step 6:
Compare the predicted probability with the actual values of the test data to find accuracy.

RESULTS:
Accuracy of digit “7”: 75.97 %
Accuracy of digit “8”: 62.73 %

LOGISTIC REGRESSION:

Step 1:
Divide the testing data into sub-test datasets for the digit “7” and digit “ 8” respectively.

Step 2:
The following values are set for training:

- Learning Rate: 0.001 (can be adjusted seeing the test data)
- Number of steps: 1000 (can be adjusted seeing the test data)

Step 3:
Define the sigmoid function. The following gives the probability of classifying an image as digit “7” (y=0) and digit “8” is given respectively:

$$P(y=0|\mathbf{x}) = \frac{1}{1+\exp\left(w_0+\sum_{i=1}^{d} w_i x_i\right)}$$

$$P(y=1|\mathbf{x}) = \frac{\exp\left(w_0+\sum_{i=1}^{d} w_i x_i\right)}{1+\exp\left(w_0+\sum_{i=1}^{d} w_i x_i\right)}$$

Step 4:
Apply Gradient Ascent Algorithm to find the predictions, that is, the probability of y given the training dataset. For that, the loss function is calculated as below:

Cost = target - predictions

Thus here, the target is the truth value dataset given for training and predictions are the values calculated using the formula above.

Step 5:
Implementing the Gradient Ascent Algorithm, predict the probability of class given the data for both the digits “7” and “8”. Before applying the algorithm for gradient ascent, calculate the gradient of the cost function by taking the dot product of transpose of train dataset for digit “7” and the weights.
Then apply the following algorithm for gradient ascent :
Iterate through the number of steps (here 1000) set above:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \eta \nabla_{\mathbf{w}^{(k)}} l(\mathbf{w})$$

where η is the learning rate which was set above (here 0.001)

Step 6:
The newly updated weights from the training dataset will now be used to find the probability of class 0 given x and probability of class 1 given x for both test dataset 7and 8 respectively.

Step 7:
Compare the predicted probability with the actual values of the test data to find accuracy.

RESULTS:
Accuracy of digit “7”: 98.75 %
Accuracy of digit “8”: 98.56 %