DEPENDABLE ARTIFICIAL INTELLIGENCE

[CSL7370]

REPORT: ASSIGNMENT 1 (BIAS)

Tejas Gaikwad

MT19AI021
Indian Institute of Technology, Jodhpur

INTRODUCTION

The assignment focuses on bias in Artificial Intelligence. The sources of bias, detection, and mitigation are the key points covered under this assignment. The assignment is divided into 2 parts, the first one focuses on evaluation metrics and detection of bias in the system. This is performed on MNIST data, and bias detection is done for '1' and '7' in the dataset. The former one focuses on neural network performance for detecting the bias and mitigating the bias via Data addition and multitasking techniques.

MATERIALS

1. Data Sets
    a. http://yann.lecun.com/exdb/mnist/
    b. https://drive.google.com/file/d/1WYb3Xonb52ZPOpyN58t0WTQPXUnwDg0U/view?usp=sharing
1. Google Colab
2. Libraries
    a. Sklearn
    b. Numpy
    c. Matplotlib
    d. OpenCV

PROCEDURE

1. Question 1
    a. access MNIST Dataset and create Training-Testing Dataset(randomizing data)
    b. Segregate 1's(6000) and 7's(500) data and labels
    c. Take 3 different 7's samples of 500
    d. Train SVM and Neural Network Model for 3 different training sets
    e. Test respective models and find confusion matrix and Prediction probability
    f. Find Mean Accuracy and standard deviation
    g. Plot ROC and Precision-Recall for different training sets

2. Question 2

a. Access the images from provided folders, create training and testing dataset from folders (1) ".\specs_train", (2) ".\specs_test", (3) ".\nonSpecs_train", (4) .\nonSpecs_test", and (5) ".\data"
b. Resizing images to 32*32
c. Training neural network of architecture [128 -- 128 -- 128 -- 64 -- 1], activation function 'sigmoid' used

d. Finding out bias with the help of the confusion matrix
e. Mitigating the bias by:

e. 1. **DATA method** (Training using more data): You may use more data for training. Use (5) “.\data” folder for extra images.

e. 2. **ALGORITHMIC method:** Alter loss function to incorporate more challenges. Use a multi-tasking approach to achieve your aim.

DATA

Dataset	URL
MNIST Data Set	http://yann.lecun.com/exdb/mnist/
Face Image Dataset	https://drive.google.com/file/d/1WYb3Xonb52ZPOpyN58t0WTQPXUnwDg0U/view?usp=sharing

DISCUSSIONS

Question 1:

Objective: To perform 2 class classifications between 1 and 7 on the MNIST dataset.

Task: Classification is performed using SVM and 5 layers Neural Network. A 5 layer neural network with architecture: [128 -- 128 -- 128 -- 64 -- 1)]

Performance Metrics:

1. Means and Standard Deviation,

2. ROC and EER(Equal Error Rate)

3. Precision-Recall Curve

Inferences from the observed results:

1. ROC Curves summarize the trade-off between the true positive rate and false-positive rate for a predictive model using different probability thresholds.
2. Precision-Recall curves summarize the trade-off between the true positive rate and the positive predictive value for a predictive model using different probability thresholds.
3. ROC curves are appropriate when the observations are balanced between each class, whereas precision-recall curves are appropriate for imbalanced datasets.

Assume you have a "positive" class called 1 and a "negative" class called 0. Y^ is your estimate of the true class label Y. Then:

- Precision=P(Y=1|Y^=1)
- Recall=Sensitivity=P(Y^=1|Y=1)
- Specificity=P(Y^=0|Y=0)

The key thing to note is that sensitivity/recall and specificity, which make up the ROC curve, are probabilities conditioned on the true class label. Therefore, they will be the same regardless of what P(Y=1) is. Precision is a probability conditioned on our estimate of the class label and will thus vary if we try our classifier in different populations with different baseline P(Y=1). However, it may be more useful in practice if we only care about one population with a known background probability and the "positive" class is much more interesting than the "negative" class. This is because it directly answers the question, "What is the probability that this is a real hit given my classifier says it is?"

So, if our question is: "How meaningful is a positive result from my classifier given the baseline probabilities of my problem?", we shall use a PR curve and If our question is, "How well can this classifier be expected to perform in general, at a variety of different baseline probabilities?", we shall go with a ROC curve.

Question 2:

Objective: To perform classification for Specs and Non Specs Faces and Mitigate the bias using 2 methods, adding more data and Multitasking approach.

Task: Perform classification using 5 layers of Neural Netowork with architecture [128 -- 128 -- 128 -- 64 -- 1)]

Performance Metrics:

For Detection:

1. ROC 2. Confusion Matrix 3. Area Under Curve(AUC)	4. Precision-Recall 5. Accuracy

Inferences from the observed results:

Bias Mitigation

Pre-Processing Algorithms:

Pre-processing algorithms are used to mitigate bias prevalent in the training data. The idea is to apply one of the following techniques for preprocessing the training data set and then apply classification algorithms for learning an appropriate classifier.

Reweighing:

Reweighing is a data preprocessing technique that recommends generating weights for the training examples in each (group, label) combination differently to ensure fairness before classification. The idea is to apply appropriate weights to different tuples in the training dataset to make the training dataset discrimination-free with respect to the sensitive attributes. Instead of reweighing, one could also apply techniques (non-discrimination constraints) such as suppression (remove sensitive attributes) or massaging the dataset — modify the labels (change the labels appropriately to remove discrimination from the training data). However, the reweighing technique is more effective than the other two mentioned earlier.

Optimized preprocessing:

The idea is to learn a probabilistic transformation that edits the features and labels in the data with group fairness, individual distortion, and data fidelity constraints and objectives.

Learning fair representations:

The idea is to find a latent representation that encodes the data well while obfuscating information about protected attributes.

Disparate impact remover:

Feature values are appropriately edited to increase group fairness while preserving rank-ordering within groups.

In-Processing Algorithms

Adversarial Debiasing:

A classifier model is learned to maximize prediction accuracy and simultaneously reduce an adversary's ability to determine the protected attribute from the predictions. This approach leads to a fair classifier as the predictions cannot carry any group discrimination information that the adversary can exploit.

Prejudice remover:

The idea is to add a discrimination-aware regularization term to the learning objective.

Post-Processing Algorithms

Equalized odds postprocessing:

The algorithm solves a linear program to find probabilities with which to change output labels to optimize equalized odds.

Calibrated equalized odds postprocessing:

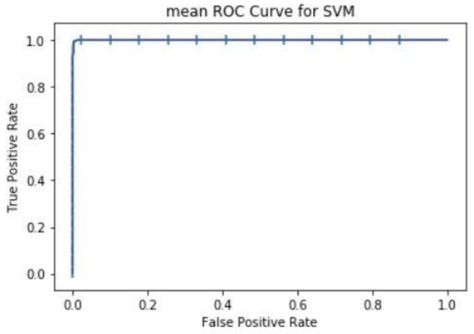
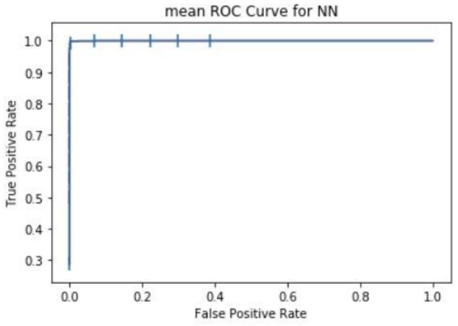
The algorithm optimizes over calibrated classifier score outputs to find probabilities with which to change output labels with an equalized odds objective.

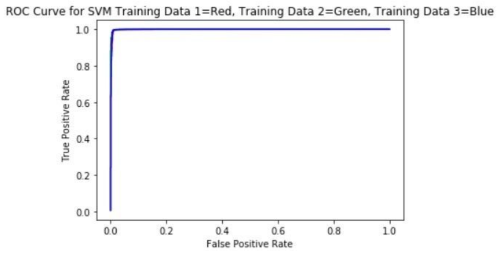
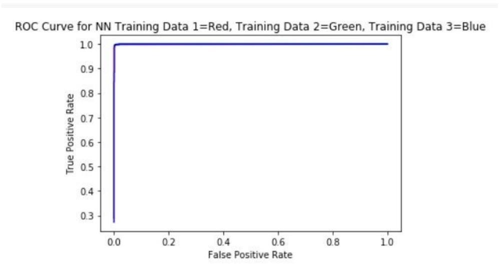
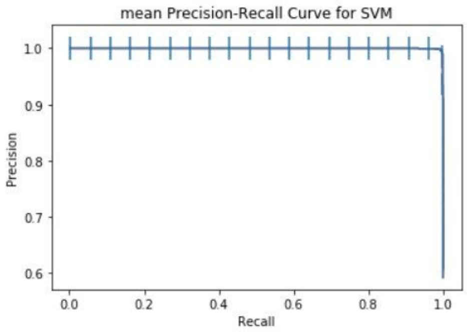
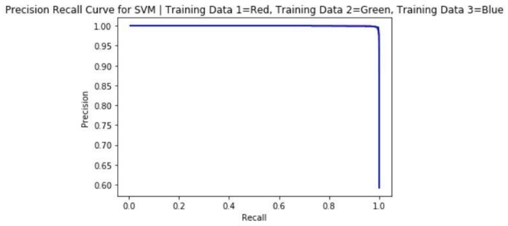
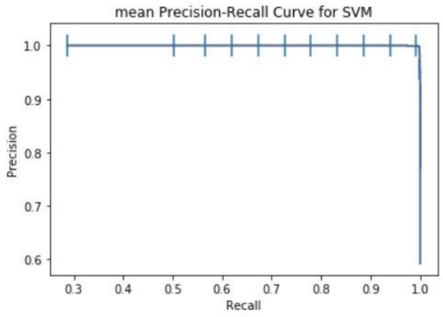
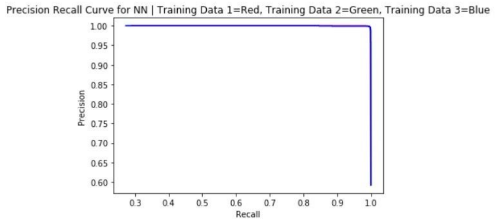
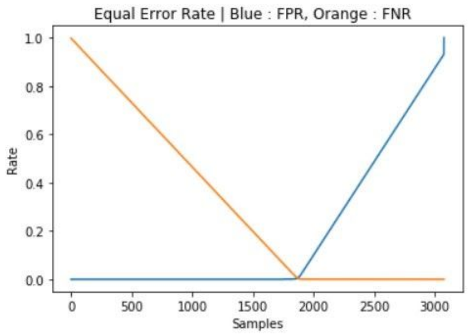
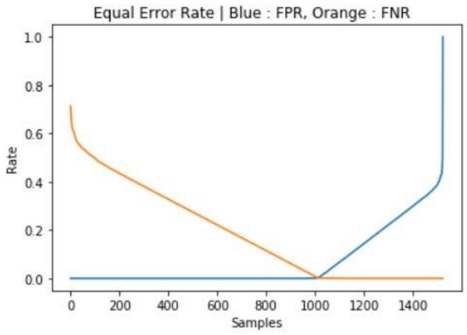
Reject option classification:

The idea is to give favorable outcomes to unprivileged groups and unfavorable outcomes to privileged groups in a confidence band around the decision boundary with the highest uncertainty.

RESULTS

QUESTION 1:

<table>
<tr><th>Sr. No.</th><th>SVM</th><th>Neural Network</th></tr>
<tr><td><b>Confusion Matrix</b></td><td><table>
<tr><td>Train Data 1</td><td>([[1877,   0],
[35, 1258]])</td></tr>
<tr><td>Train Data 2</td><td>([[1876,   1],
[38, 255]])</td></tr>
<tr><td>Train Data 3</td><td>([[1877,   0],
[30, 1263]])</td></tr>
</table></td><td><table>
<tr><td>Train Data 1</td><td>([[1876, 1],
[16, 1277]])</td></tr>
<tr><td>Train Data 2</td><td>([[1876,   1],
[22, 1271]])</td></tr>
<tr><td>Train Data 3</td><td>([[1876,   1],
[17, 1276]])</td></tr>
</table></td></tr>
<tr><td><b>Accuracy and Deviation</b></td><td>Accuracy and Deviation for Training Data
1)98.9+ -0.0066666666666674814
2)98.77+ -0.13666666666668448
3)99.05 + 0.14333333333331666
=============================
Accuracy for SVM : 98.91%
=============================</td><td>Accuracy and Deviation for Training Data
1)98.9+-0.48666666666665037
2)98.77 + -0.61666666666666
3)99.05 + -0.33666666666666589
=============================
Accuracy for SVM : 99.39 %
=============================</td></tr>
<tr><td><b>Receiver Operating Characteristics (ROC)</b></td><td>
</td><td>
</td></tr>
</table>

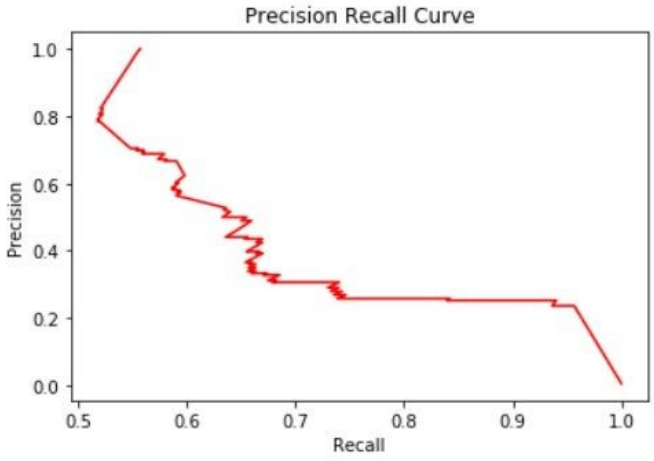
	 	
Precision-Recall	  	 
Equal Error Rate	 	

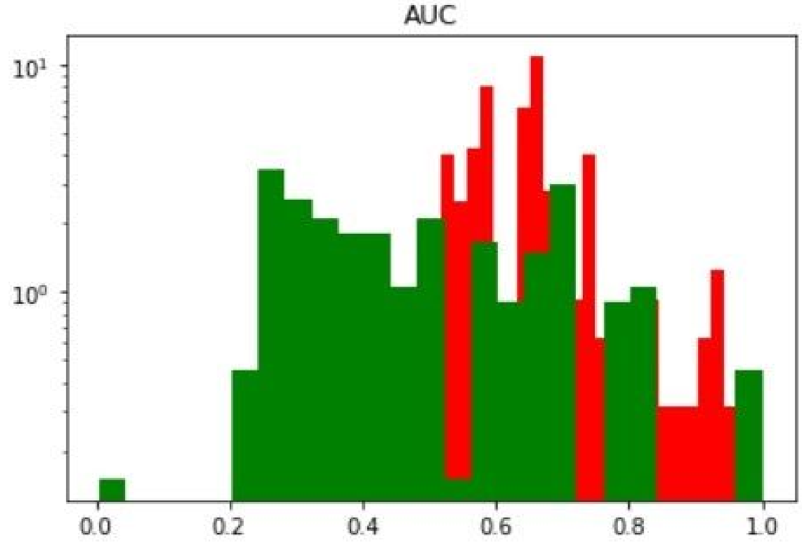
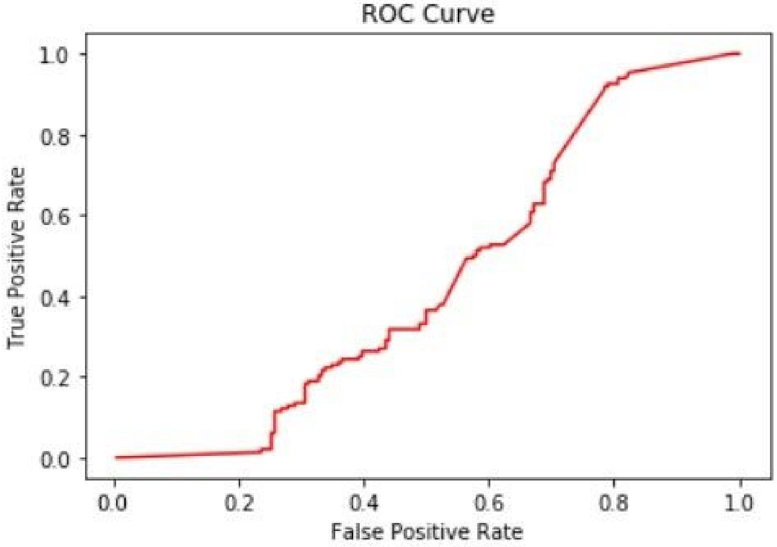
QUESTION 2:

Accuracy:

41.916 %

Evaluation Metrics to detect Bias:

Metrics	**Result**
Precision Recall	
Confusion Matrix	[[36 112] [82 104]]

Area Under Curve(AUC)	
Receiver Operating Characteristics	

Bias Mitigation:

Before adding New Data	After adding New Data
[[36 112] [82 104]]	[[106 42] [102 84]]
Accuracy: 41.916 %	Accuracy:56.886 %

Conclusion:

I have learned the basic idea behind a bias system, ways of detection the bias, the metrics to be used when to measure the bias and the ways to mitigate the bias.