Minior Project Report

M Purna Chandra Sekhar, *B19CSE046* Nunna Radhasyam, *B19CSE060*

Abstract

This mini-project is based on Image classification. The goal of this mini-project is to make atleast three reasonably good classifiers to classify the images between two classes, which are ”wearing mask” and ”not wearing mask”.

I. Introduction

IN this report for our mini-project we are going to discuss about detailed working of our face mask detector/classifier. We are also going to talk about the machine learning pipeline. We have used the images which were provided for the training, validation and testing purposes. If the person is wearing mask, we have given them as class ”1” and ”0” otherwise.

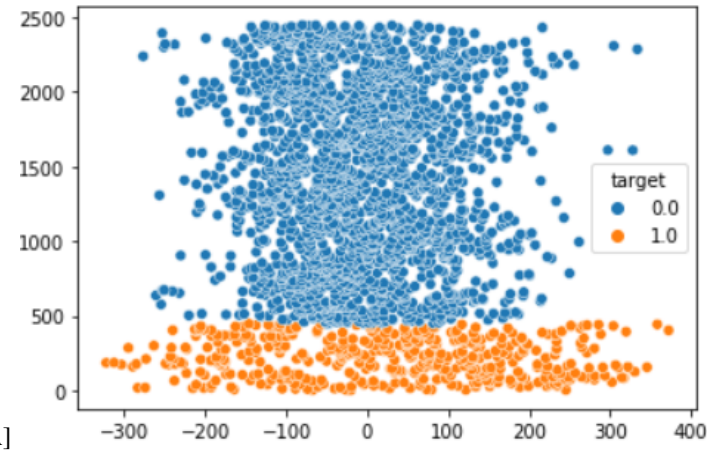
II. STEPS IN OUR PIPELINE

A. Data Analysis and Preprocessing

1) The starting step in the machine learning pipeline is preprocessing the data. During the preprocessing, The images of masked(447 images are taken) and non masked(2000 images are taken) are taken from the data set provided.
2) We took less images as it would be convenient for us to run the colab notebook. The Non masked and masked images are read into different lists Nonmaskedimages , Maskedimages respectively.
3) As images are of different different pixels we had resized then into (128,128) pixels this results into 128*128*3 =49152 columns of data for each image(*3 is done as it is a RGB image).
4) This data of each image is added into lists by using suitable loops into masked and Nonmasked lists which contain masked and non masked images data respectively.
5) We grouped masked data to class 1 and Nonmasked data to class 0.
6) Then we have created two dataframes and for masked and Nonmasked then we finally merged them into a single dataframe called ”df”.
7) We also deleted the intial lists and dataframes of the separate classes as they consuming a lot of memory, and thereby causing out environment to crash.
8) The shape of final dataframe of the data, ”df” is (2447,49153).

B. Dimensionality Reduction

1) Principal Component Analysis(PCA)
   - It is a method that uses simple matrix operations from linear algebra and statistics to calculate a projection of the original data into the same number or fewer dimensions.
   - We have used this method using 90 percent of the variation from the original data. The number of components which we finally obtained were 82.
   - This saves us a lot of computation time as the the dimensionality was reduced to 82 whereas before it was 49513. Also we have 90percent of the variation so this was infact very beneficial

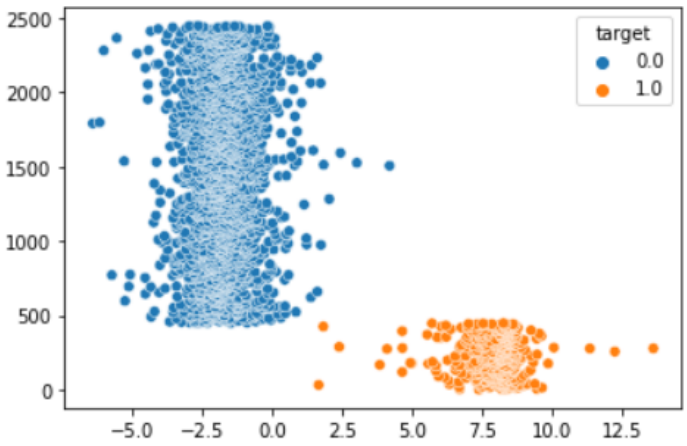


[Fig 1.PCA]

2) Linear Discriminant Analysis(LDA)

M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see http://www.michaelshell.org/contact.html).
J. Doe and J. Doe are with Anonymous University.
Manuscript received April 19, 2005; revised August 26, 2015.

- Linear Discriminant Analysis or Normal Discriminant Analysis or Discriminant Function Analysis is a dimensionality reduction technique which is commonly used for the supervised classification problems.
- It is used to reduce the dimensions in such a way that the maximum class separatibility is obtained
- This usually gives n-1 dimensions where n is he number of features



[Fig 1.PCA]

C. Classification

1) Multi Layer Perceptron Classifier(MLP) Classifier
   - This is very robust machine learning algorith used in various types of problems for classification and regression alike.
   - We have given the following hyper parameters - activation='relu',solver='adam',hidden$_layer_sizes = (100,), alpha = 0.0001, batch_size = 250, learning_rate_init = 0.001, random_state = 42, warm_start = FalseWehaveusedtrialanderrorandsom$
2) Decision Tree Classifier
   - A Decision Tree is a simple supervised learning method used for classification and regression.
   - The goal is to create a model that predicts the value of a target variable by learning simple decision rules (if-else) inferred from the data features.
   - We have used the following hyper paramters - max$_depth = 7, min_samples_split = 0.08, max_leaf_nodes = 15Wehaveusedthesepara$
3) Support Vector Machine
   - The Support Vector Machine is a simple algorithm for classification and regression tasks. It can provide high accuracy with less computation power very fast.
   - We have used LinearSVC. It turned out that setting the regularisation parameter C=0.0001 improves the quality of the prediction and reduces overfitting.
   - We found this hyperparameter after some trials

III. Hypotheses

1) We expect the decision tree to overfit generally, but we have given the hyper parameters in a way that the risk of overfiting is low
2) MLP and SVMs generally give good accuracy and we expect them to outperform Decision tree

IV. Observations

1) All the classifiers performed best on the dataset which was obtained from doing LDA.
2) For MLP
   - without dimesninality reduction
     - cross val scores - [0.88163265, 0.91020408, 0.95705521, 0.90797546, 0.81799591]
     - Accuracy - 0.954
     - F1 score - 0.872
     - AUC - 0.901
   - with PCA
     - cross val scores - [0.93877551, 0.91020408, 0.90797546, 0.93865031, 0.95092025]
     - Accuracy - 0.951
     - F1 score - 0.865
     - AUC - 0.918

- with LDA
  - cross val scores - [0.99795918, 1. , 0.99386503, 0.99795501, 0.99795501]
  - Accuracy - 0.996
  - F1 score - 0.989
  - AUC - 0.998

3) For Decision tree
   - without dimesionality reduction
     - cross val scores - [0.91632653, 0.90204082, 0.8997955 , 0.90593047, 0.94478528]
     - Accuracy - 0.92
     - F1 score - 0.773
   - AUC - 0.842
   - with PCA
     - cross val scores - [0.86938776, 0.83469388, 0.83640082, 0.8609407 , 0.85685072]
     - Accuracy - 0.89
     - F1 score - 0.654
     - AUC - 0.767
   - with LDA
     - cross val scores - [0.99795918, 0.99795918, 0.98977505, 0.99591002, 0.99795501]
     - Accuracy - 0.996
     - F1 score - 0.989
     - AUC - 0.998
5) For AdaBoost
   - cross val score[0.92892157, 0.93382353, 0.95705521])
   - Accuracy - 0.969
   - F1 score - 0.911
   - AUC - 0.929
   - cross val score - [0.99877451, 0.99387255, 0.99877301]
   - Accuracy - 0.996
   - F1 score - 0.989
   - AUC - 0.998

V. Conclusion

1) The classifiers were working best for the dataset obtained from LDA. this was expected as it increases the class separatibity
2) MLP was performing best

VI. Contribution

1) M Purna Chandra Sekhar Reddy(B19CSE046) - Model Selection, Report
2) Nunna Radhasyam(B19CSE060) - Data Analysis, Pre-processing, Report

VII. References

- Dataset - https://drive.google.com/drive/folders/1d9vG30l1z6pXL7K4-NeFRHesQOif6yjW?usp=sharing
- https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- https://scikit-learn.org/stable/modules/generated/sklearn.neural$_n etwork.MLPClassifier.htmlhttps://scikit-learn.org/stable/m$
- https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
- https://docs.opencv.org/master/d4/da8/group$_imgcodecs.html$