**Machine Learning**

**(BITS F464)**

**Assignment 3**

**Naïve Bayes Classifier**

# Training using Naïve Bayes Classifier

We have 451 different images in the training data. Each image comprises of 60X70 Pixels.

The training data comprises of 150 test images and for each image we calculate the likelihood of yes and likelihood of no. If likelihood of yes is greater we classify it as a face else classify it as not a face.

Confusion Matrix

|  |  |  |
| --- | --- | --- |
| **Total =150** | **Predicted YES** | **Predicted NO** |
| **Actual YES** | **63** | **10** |
| **Actual NO** | **7** | **70** |

|  |  |
| --- | --- |
| True Positive | 63 |
| True Negative | 70 |
| False Positive | 7 |
| False Negative | 10 |

**Accuracy:**

Test set : 88.6667%

EXAMPLE OF FALSE POSITIVE

#

# ##

# #

# #

## #

# #

# #

# #

# #

# #

# #

# #

# #

# # # # # #

# #

# #

# #

# #

# #

# #

# #

# #

# #

# #

# #

# #

# # #######################

# # #### #

# # ### #####

# # #####

# # ###

# # ###

# # # #

# # # ##

# # # ##

# # # #

# # # ##

# # #

# # #

# ### #

# # # #

# # #

# #### # ## #

## ### # ###

# ##

# # # #

# # # # #

# # # # #

# # # # # # # # # #

# ### ### #

#

# #

# # ## ## #

# ###### ### #

# #

# #

### # # #########

# # # #

# ## # #

# ### ## ####

# ############# #####

# ##

## ## ###

###

### Example of False Negative

####

#### #

# #### ####

# ####

# # #

###### ## #

# # ##

###### ### #

# # #

# # #

# ### # #

# # ## #

# # ## ###### # #

# # ## ## # #

# # # # #

# # # # #

# # # # #

# # # # #

# # # # #

# # ### # # #

# # # # # #

# # # # # #

# # # # # #

# # ## # ### # # #

# # # # # # # # #

# # # # # # ### #

## # # ### # # #

### # ### # # # #

# # # # # # # # # # #

# # # # # #

# # # # # #

# ### # # # #

# # # # # #

# # # # # #

# # # # # #

# # # # #

# # # # #

# # # # #

# # ## # #

# # # #

# # # # #

# # ## #

# # # #

# # # #

# ### # #

# ####### #

# ## #

## # # #

# ####### # #

###### # # #

#### # #

## # # # #

# # # # # #

# ## # # #

# # # # # ##

# # # # # ###

# # # # # ##

## # # # # ##

## # ## # ###

### # # #

# # ## #

# # ######### #

# #

**CODE:**

#include <iostream>

#include <bits/stdc++.h>

using namespace std; char sttemp[400];

int face\_count = 0,total\_face = 0,total\_nonface = 0; int trainface[10000][80][80]; int traindo[10000];

int testingface[10000][80][80]; int testdo[10000]; int confusion\_matrix[2][2];

double prob[4][80][80],probofface,probofnonface;

void find\_probabilty(int x,int y){

int ppfy = 0,pnfc = 0,ppfn = 0,pnfn = 0;

for(int i = 0; i<face\_count;i++){

if(trainface[i][x][y] == 1 && traindo[i] == 1){

ppfy++;

}

if(trainface[i][x][y] == 1 && traindo[i] == 0){

ppfn++;

}

if(trainface[i][x][y] == 0 && traindo[i] == 1){

pnfc++;

}

if(trainface[i][x][y] == 0 && traindo[i] == 0){

pnfn++;

}

}

prob[0][x][y] = (double)ppfy/(double)(total\_face+1); prob[0][x][y] = log(prob[0][x][y]);

prob[1][x][y] = (double)ppfn/(double)(total\_nonface+1); prob[1][x][y] = log(prob[1][x][y]);

prob[2][x][y] = (double)pnfc/(double)(total\_face+1); prob[2][x][y] = log(prob[2][x][y]);

prob[3][x][y] = (double)pnfn/(double)(total\_nonface+1); prob[3][x][y] = log(prob[3][x][y]); }

int test(int face){ //TEST Function double lf = probofface,lnf = probofnonface;

int temp = 0;

for(int i = 1; i<=70;i++) for(int j = 1;j<=59;j++){ if(testingface[face][i][j] == 1){ lf+=prob[0][i][j];

lnf+=prob[1][i][j];

}

else{ lf+=prob[2][i][j];

lnf+=prob[3][i][j];

} if(lf>lnf) temp = 1;

else if(lf <lnf)

temp = 0;

}

return temp;

}

int main(){

//READING TRAINDATALABELS FILE AND STORING IN traindo

ARRAY

FILE \*inputfile = fopen("facedatatrainlabels","r+"); int ct = 0;

while(fscanf(inputfile, "%[^\n]s",sttemp)!=EOF){

int temp = int(sttemp[0]-'0'); traindo[ct] = temp;

if(temp)total\_face++;

else

total\_nonface++; ct++;

fgetc(inputfile);

}

ct = 1,face\_count = 0;

inputfile = fopen("facedatatrain","r+");

while(fscanf(inputfile, "%[^\n]s",sttemp)!=EOF){

if(ct%70 == 0){ ct = 0; face\_count++;

}

int len = strlen(sttemp),j = 0;

for(int i = 0; i<len;i++){ if(sttemp[i] == '#')

trainface[face\_count][ct][i] = 1;

} ct++;

fgetc(inputfile);

}

//probability for each pixel for(int i = 1; i<=70;i++) for(int j = 1;j<=59;j++)

find\_probabilty(i,j);

inputfile = fopen("facedatatestlabels","r+"); ct = 0;

//reading testing data

while(fscanf(inputfile, "%[^\n]s",sttemp)!=EOF){ int temp = int(sttemp[0]-'0');

testdo[ct] = temp;

ct++;

fgetc(inputfile);

}

ct = 1;

inputfile = fopen("facedatatest","r+");

face\_count = 0;

while(fscanf(inputfile, "%[^\n]s",sttemp)!=EOF){

if(ct%70 == 0){ ct = 0;

face\_count++;

}

int len = strlen(sttemp),j = 0;

for(int i = 0; i<len;i++){

if(sttemp[i] == '#')

testingface[face\_count][ct][i] = 1;

} ct++;

fgetc(inputfile);

}

//testing

probofface = (double)total\_face/(double)(total\_face+total\_nonface+1); probofnonface = (double)total\_nonface/(double)(total\_nonface+total\_face+1);

ofstream outputFile;

outputFile.open("BayesClassifierOutput.txt");

outputFile << setw(3) << "#" << setw(16) << "Given Class" << setw(31) << right

<< "Predicted Class" << endl;

outputFile << "--------------------------------------------------" << endl; int cor = 0;

for(int k = 0; k<face\_count;k++){ int x = test(k);

outputFile << setw(3) << k+1 << setw(16) << x;

if ((x == 1 &&testdo[k] == 1)||(x == 0 && testdo[k] == 0) )

{

cor++;

outputFile << " ------------ ";

} else

outputFile << " xxxxxxxxxxxx ";

outputFile << testdo[k] << endl;

if(x == 0 && testdo[k] == 0) confusion\_matrix[0][0]++; if(x == 1 && testdo[k] == 1)

confusion\_matrix[1][1]++;

if(x == 0 && testdo[k] == 1) confusion\_matrix[1][0]++;

if(x == 1 && testdo[k] == 0)

confusion\_matrix[0][1]++;

}

cout<<"Confusion Matrix"<<endl<<" "<<confusion\_matrix[0][0]<<" "<<confusion\_matrix[0][1]<<endl<<" "<<confusion\_matrix[1][0]<<"

"<<confusion\_matrix[1][1]<<endl;

double acc = (double)cor/(double)face\_count\*100;

outputFile << "--------------------------------------------------" << endl; outputFile << "Total number of instances in test data = " << face\_count << endl; outputFile << "Number of correctly predicted instances = " << cor << endl; outputFile << "Accuracy = " << acc <<"%"<< endl; outputFile.close();

cout<<"Accuracy is: " <<acc<<"%"<<endl;

cout<<"(Check BayesClassifierOutput.txt for more details)"<<endl;

}