|  |  |  |
| --- | --- | --- |
| **Discrete Mathematics** | Section |  |
| Student number |  |
| **HW6 – Spam filter** | Name | Hong, Gil Dong |

*If your explanation is less informative and insufficient, then you may not get any points.*

*Also, you should provide discussion, otherwise you will get penalty.*

* General information

|  |  |
| --- | --- |
| Item | Your answer |
| The number of lines in your code. | 190 |
| The number of functions in your code. | 0 |

* Functions 사용하지 않았음.

|  |  |
| --- | --- |
| Function name | Function Description |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* Screenshot of your program running

텍스트이(가) 표시된 사진

자동 생성된 설명

* Results
  + Probability and predicted label (spam or non-spam) for different threshold (T) (0.6, 0.7, 0.8, 0.9 and 0.95)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| num  True | True Label | T = 0.6 | | T = 0.7 | | T = 0.8 | | T = 0.9 | | T = 0.95 | |
|  | predicted  label |  | predicted  label |  | predicted  label |  | predicted  label |  | predicted  label |
| s01 | Spam | 0.394701 | Non-spam | 0.394701 | Non-spam | 0.394701 | Non-spam | 0.394701 | Non-spam | 0.394701 | Non-spam |
| s02 | Spam | 5.78635e-007 | Non-spam | 5.78635e-007 | Non-spam | 5.78635e-007 | Non-spam | 5.78635e-007 | Non-spam | 5.78635e-007 | Non-spam |
| s03 | Spam | 0.00191421 | Non-spam | 0.00191421 | Non-spam | 0.00191421 | Non-spam | 0.00191421 | Non-spam | 0.00191421 | Non-spam |
| s04 | Spam | 1.64974e-007 | Non-spam | 1.64974e-007 | Non-spam | 1.64974e-007 | Non-spam | 1.64974e-007 | Non-spam | 1.64974e-007 | Non-spam |
| s05 | Spam | 8.50748e-018 | Non-spam | 8.50748e-018 | Non-spam | 8.50748e-018 | Non-spam | 8.50748e-018 | Non-spam | 8.50748e-018 | Non-spam |
| s06 | Spam | 0.0298705 | Non-spam | 0.0298705 | Non-spam | 0.0298705 | Non-spam | 0.0298705 | Non-spam | 0.0298705 | Non-spam |
| s07 | Spam | 9.96454e-005 | Non-spam | 9.96454e-005 | Non-spam | 9.96454e-005 | Non-spam | 9.96454e-005 | Non-spam | 9.96454e-005 | Non-spam |
| s08 | Spam | 3.87732e-025 | Non-spam | 3.87732e-025 | Non-spam | 3.87732e-025 | Non-spam | 3.87732e-025 | Non-spam | 3.87732e-025 | Non-spam |
| s09 | Spam | 5.05687e-014 | Non-spam | 5.05687e-014 | Non-spam | 5.05687e-014 | Non-spam | 5.05687e-014 | Non-spam | 5.05687e-014 | Non-spam |
| s10 | Spam | 6.31595e-010 | Non-spam | 6.31595e-010 | Non-spam | 6.31595e-010 | Non-spam | 6.31595e-010 | Non-spam | 6.31595e-010 | Non-spam |
| s11 | Spam | 8.52419e-016 | spam | 8.52419e-016 | Non-spam | 8.52419e-016 | Non-spam | 8.52419e-016 | Non-spam | 8.52419e-016 | Non-spam |
| s12 | Spam | 1.60552e-019 | Non-spam | 1.60552e-019 | Non-spam | 1.60552e-019 | Non-spam | 1.60552e-019 | Non-spam | 1.60552e-019 | Non-spam |
| s13 | Spam | 2.07774e-017 | Non-spam | 2.07774e-017 | Non-spam | 2.07774e-017 | Non-spam | 2.07774e-017 | Non-spam | 2.07774e-017 | Non-spam |
| s14 | Spam | 2.63461e-016 | Non-spam | 2.63461e-016 | Non-spam | 2.63461e-016 | Non-spam | 2.63461e-016 | Non-spam | 2.63461e-016 | Non-spam |
| s15 | Spam | 3.12971e-013 | Non-spam | 3.12971e-013 | Non-spam | 3.12971e-013 | Non-spam | 3.12971e-013 | Non-spam | 3.12971e-013 | Non-spam |
| s16 | Spam | 1 | spam | 1 | spam | 1 | spam | 1 | spam | 1 | Non-spam |
| s17 | Spam | 0.137989 | Non-spam | 0.137989 | Non-spam | 0.137989 | Non-spam | 0.137989 | Non-spam | 0.137989 | Non-spam |
| s18 | Spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam |
| s19 | Spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam |
| s20 | Spam | 8.71929e-020 | Non-spam | 8.71929e-020 | Non-spam | 8.71929e-020 | Non-spam | 8.71929e-020 | Non-spam | 8.71929e-020 | Non-spam |
| h01 | Non-spam | 0.394701 | Non-spam | 0.394701 | Non-spam | 0.394701 | Non-spam | 0.394701 | Non-spam | 0.394701 | Non-spam |
| h02 | Non-spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam |
| h03 | Non-spam | 1.32952e-026 | Non-spam | 1.32952e-026 |  | 1.32952e-026 |  | 1.32952e-026 |  | 1.32952e-026 |  |
| h04 | Non-spam | 3.3284e-013 | Non-spam | 3.3284e-013 |  | 3.3284e-013 |  | 3.3284e-013 |  | 3.3284e-013 |  |
| h05 | Non-spam | 1.34502e-030 | Non-spam | 1.34502e-030 |  | 1.34502e-030 |  | 1.34502e-030 |  | 1.34502e-030 |  |
| h06 | Non-spam | 4.32898e-028 | Non-spam | 4.32898e-028 |  | 4.32898e-028 |  | 4.32898e-028 |  | 4.32898e-028 |  |
| h07 | Non-spam | 1.50685e-032 | Non-spam | 1.50685e-032 |  | 1.50685e-032 |  | 1.50685e-032 |  | 1.50685e-032 |  |
| h08 | Non-spam | 0 | Non-spam | 0 |  | 0 |  | 0 |  | 0 |  |
| h09 | Non-spam | 4.31258e-029 | Non-spam | 4.31258e-029 |  | 4.31258e-029 |  | 4.31258e-029 |  | 4.31258e-029 |  |
| h10 | Non-spam | 7.07622e-011 | Non-spam | 7.07622e-011 |  | 7.07622e-011 |  | 7.07622e-011 |  | 7.07622e-011 |  |
| h11 | Non-spam | 0 | Non-spam | 0 |  | 0 |  | 0 |  | 0 |  |
| h12 | Non-spam | 2.3285e-035 | Non-spam | 2.3285e-035 |  | 2.3285e-035 |  | 2.3285e-035 |  | 2.3285e-035 |  |
| h13 | Non-spam | 1.22784e-021 | Non-spam | 1.22784e-021 |  | 1.22784e-021 |  | 1.22784e-021 |  | 1.22784e-021 |  |
| h14 | Non-spam | 0 | Non-spam | 0 |  | 0 |  | 0 |  | 0 |  |
| h15 | Non-spam | 0.000111447 | Non-spam | 0.000111447 | Non- spam | 0.000111447 | Non- spam | 0.000111447 | Non- spam | 0.000111447 | Non- spam |
| h16 | Non-spam | 1.81876e-019 | Non-spam | 1.81876e-019 | Non-spam | 1.81876e-019 | Non-spam | 1.81876e-019 | Non-spam | 1.81876e-019 | Non-spam |
| h17 | Non-spam | 4.98728e-017 | Non-spam | 4.98728e-017 | Non-spam | 4.98728e-017 | Non-spam | 4.98728e-017 | Non-spam | 4.98728e-017 | Non-spam |
| h18 | Non-spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam |
| h19 | Non-spam | 1.93032e-018 | Non-spam | 1.93032e-018 | Non-spam | 1.93032e-018 | Non-spam | 1.93032e-018 | Non-spam | 1.93032e-018 | Non-spam |
| h20 | Non-spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam | 0 | Non-spam |
| Accuracy (%) | |  | |  | |  | |  | |  | |

※ Accuracy is calculated from the following equation.

* Discussion (your interpretation of the results and possible strategy to improve the algorithm)

□ Codes

// Put code here, and you should also submit your original executable C code.

// If C code does not run, then No point will be given.

#include <iostream>

#include <fstream>

#include <sstream>

#include <vector>

#include <string>

#include <algorithm>

using namespace std;

int main(){

    ifstream testh;

    ifstream tests;

    ifstream trainh;

    ifstream trains;

    testh.open("csv/test/dataset\_ham\_test20.csv");

    tests.open("csv/test/dataset\_spam\_test20.csv");

    trainh.open("csv/train/dataset\_ham\_train100.csv");

    trains.open("csv/train/dataset\_spam\_train100.csv");

    vector<vector<string>> ateh;

    vector<string> teh;

    vector<vector<string>> ates;

    vector<string> tes;

    vector<vector<string>> atrh;

    vector<string> trh;

    vector<vector<string>> atrs;

    vector<string> trs;

    string chars = "'~!@#$%^&\*()-<>?/.,:\"";

    string s;

    string t;

    while(getline(testh,s)){

        for (char c: chars) {

            s.erase(std::remove(s.begin(), s.end(), c), s.end());

        }

        istringstream ss(s);

        while (getline(ss, t, ' ')){

            if(t == "")continue;

            for(int i=0; i<t.length(); i++){

                if(t.find("Subject") == -1){

                    break;

                }else if(t.find("Subject",i)<=i){

                    ateh.push\_back(teh);

                    teh.clear();

                }

            }

            teh.push\_back(t);

        }

    }

    while(getline(tests,s)){

        for (char c: chars) {

            s.erase(std::remove(s.begin(), s.end(), c), s.end());

        }

        istringstream ss(s);

        while (getline(ss, t, ' ')){

            if(t == "")continue;

            for(int i=0; i<t.length(); i++){

                if(t.find("Subject") == -1){

                    break;

                }else if(t.find("Subject",i)<=i){

                    ates.push\_back(tes);

                    tes.clear();

                }

            }

            tes.push\_back(t);

        }

    }

    while(getline(trainh,s)){

        for (char c: chars) {

            s.erase(std::remove(s.begin(), s.end(), c), s.end());

        }

        istringstream ss(s);

        while (getline(ss, t, ' ')){

            if(t == "")continue;

            trh.push\_back(t);

        }

        atrh.push\_back(trh);

        trh.clear();

    }

    while(getline(trains,s)){

        for (char c: chars) {

            s.erase(std::remove(s.begin(), s.end(), c), s.end());

        }

        istringstream ss(s);

        while (getline(ss, t, ' ')){

            if(t == "")continue;

            trs.push\_back(t);

        }

        atrs.push\_back(trs);

        trs.clear();

    }

    int bunja = 0;

    int bunmo = 1;

    float per1;

    float per2;

    float temp;

    float k = 1;

    for(int i = 0; i < ateh.size(); i++){

        for(int j = 0; j < ateh[i].size(); j++){

            for(int a = 0; a < atrs.size(); a++){

                for(int b = 0; b < atrs[a].size(); b++){

                        if(ateh[i][j] == atrs[a][b]){

                            bunja ++;

                        }

                    bunmo ++;

                }

            }

            per1 = (float)bunja / (float)bunmo;

            bunja = 0;

            bunmo = 1;

            for(int a = 0; a < atrh.size(); a++){

                for(int b = 0; b < atrh[a].size(); b++){

                        if(ateh[i][j] == atrh[a][b]){

                            bunja ++;

                        }

                    bunmo ++;

                }

            }

            per2 = (float)bunja / (float)bunmo;

            temp = per1 + per2;

            if(temp == 0)temp = 1;

            if(per1/temp == 0){

                continue;

            }

            k \*= (per1/temp);

        }

        cout << "Probability for r" << i<<" = "<<k << "\n";

        cout << "------Finding Probability Start-----\n------Finding Probability End-------\n";

        k = 1;

    }

    bunja = 0;

            bunmo = 1;

    for(int i = 0; i < ates.size(); i++){

        for(int j = 0; j < ates[i].size(); j++){

            for(int a = 0; a < atrs.size(); a++){

                for(int b = 0; b < atrs[a].size(); b++){

                        if(ates[i][j] == atrs[a][b]){

                            bunja ++;

                        }

                    bunmo ++;

                }

            }

            per1 = (float)bunja / (float)bunmo;

            bunja = 0;

            bunmo = 1;

            for(int a = 0; a < atrh.size(); a++){

                for(int b = 0; b < atrh[a].size(); b++){

                        if(ates[i][j] == atrh[a][b]){

                            bunja ++;

                        }

                    bunmo ++;

                }

            }

            per2 = (float)bunja / (float)bunmo;

            temp = per1 + per2;

            if(temp == 0)temp = 1;

            if(per1/temp == 0){

                continue;

            }

            k \*= (per1/temp);

        }

        cout << "Probability for r" << i<<" = "<<k << "\n";

        cout << "------Finding Probability Start-----\n------Finding Probability End-------\n";

        k = 1;

    }

}