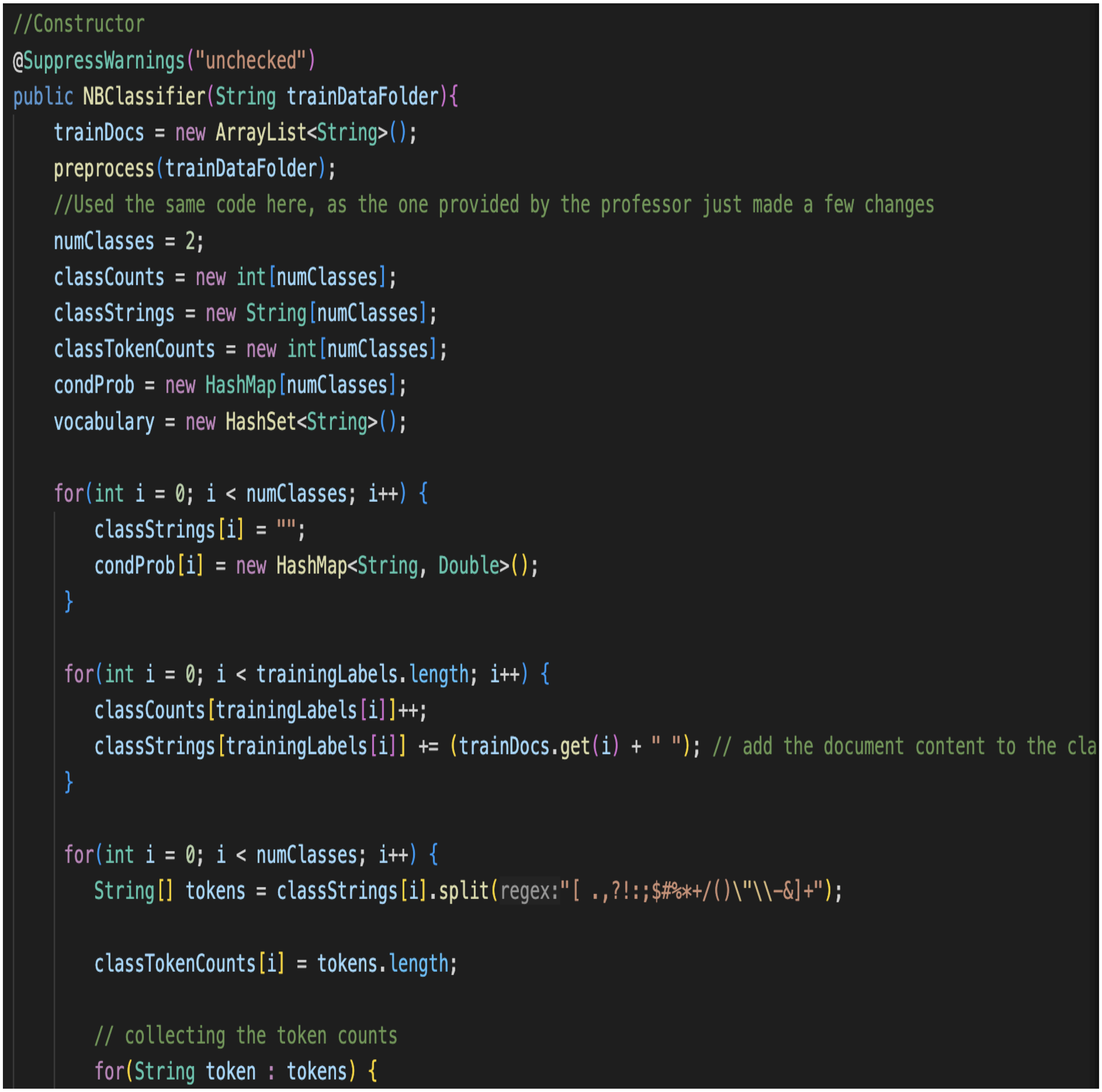
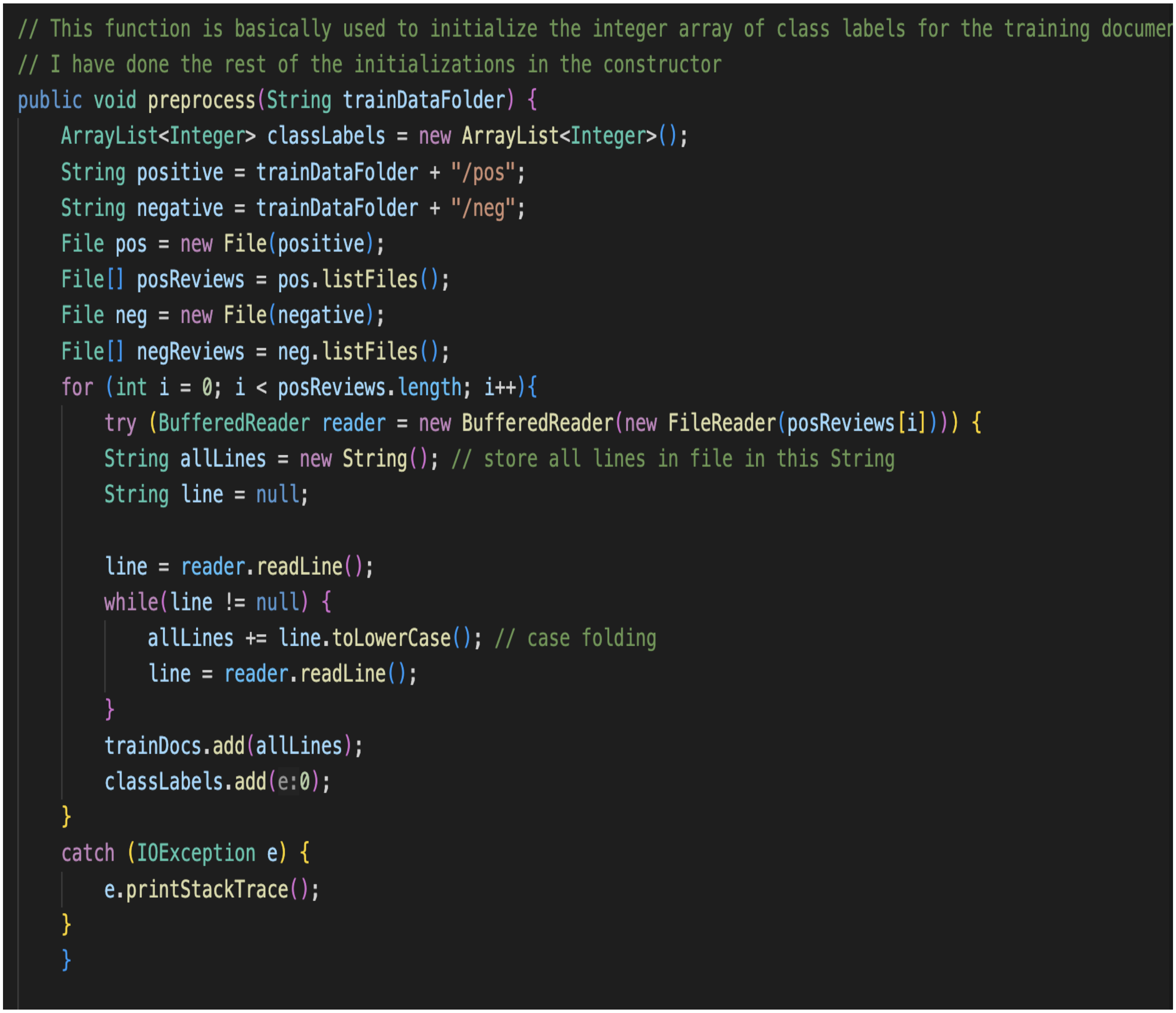
Lab 4

Name: Srivenkatesh Nair
Student ID: 378000329
Username: sn6711
ISTE.612

Task 1 (40 points): Building the Naïve Bayes Classifier

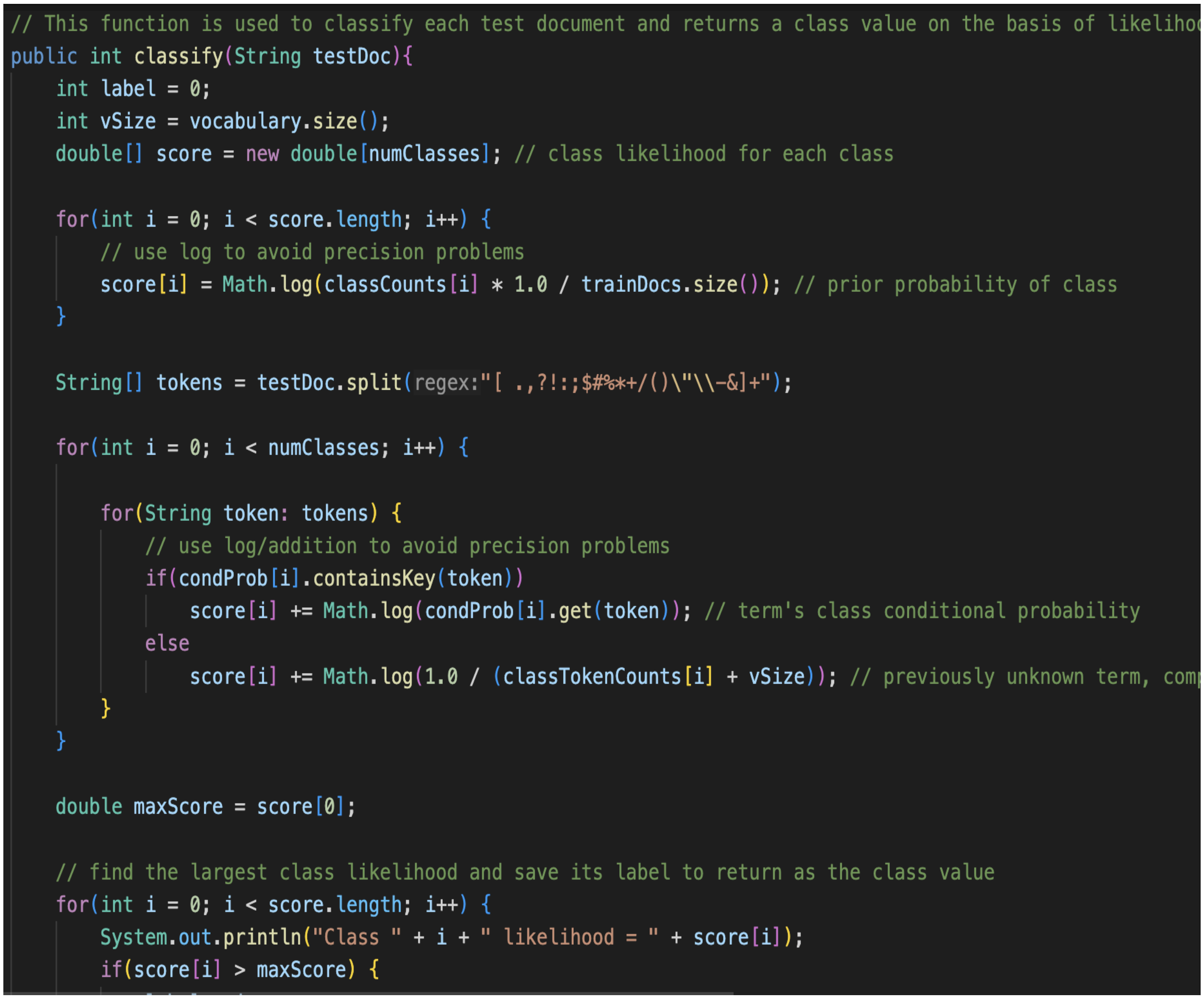


```
//Constructor
@SuppressWarnings("unchecked")
public NBClassifier(String trainDataFolder){
    trainDocs = new ArrayList<String>();
    preprocess(trainDataFolder);
    //Used the same code here, as the one provided by the professor just made a few changes
    numClasses = 2;
    classCounts = new int[numClasses];
    classStrings = new String[numClasses];
    classTokenCounts = new int[numClasses];
    condProb = new HashMap[numClasses];
    vocabulary = new HashSet<String>();

    for(int i = 0; i < numClasses; i++) {
        classStrings[i] = "";
        condProb[i] = new HashMap<String, Double>();
    }

    for(int i = 0; i < trainingLabels.length; i++) {
        classCounts[trainingLabels[i]]++;
        classStrings[trainingLabels[i]] += (trainDocs.get(i) + " "); // add the document content to the cla
    }

    for(int i = 0; i < numClasses; i++) {
        String[] tokens = classStrings[i].split(regex:"[ .,?!:;$#%*+/()\"\\-&]+");

        classTokenCounts[i] = tokens.length;

        // collecting the token counts
        for(String token : tokens) {
```



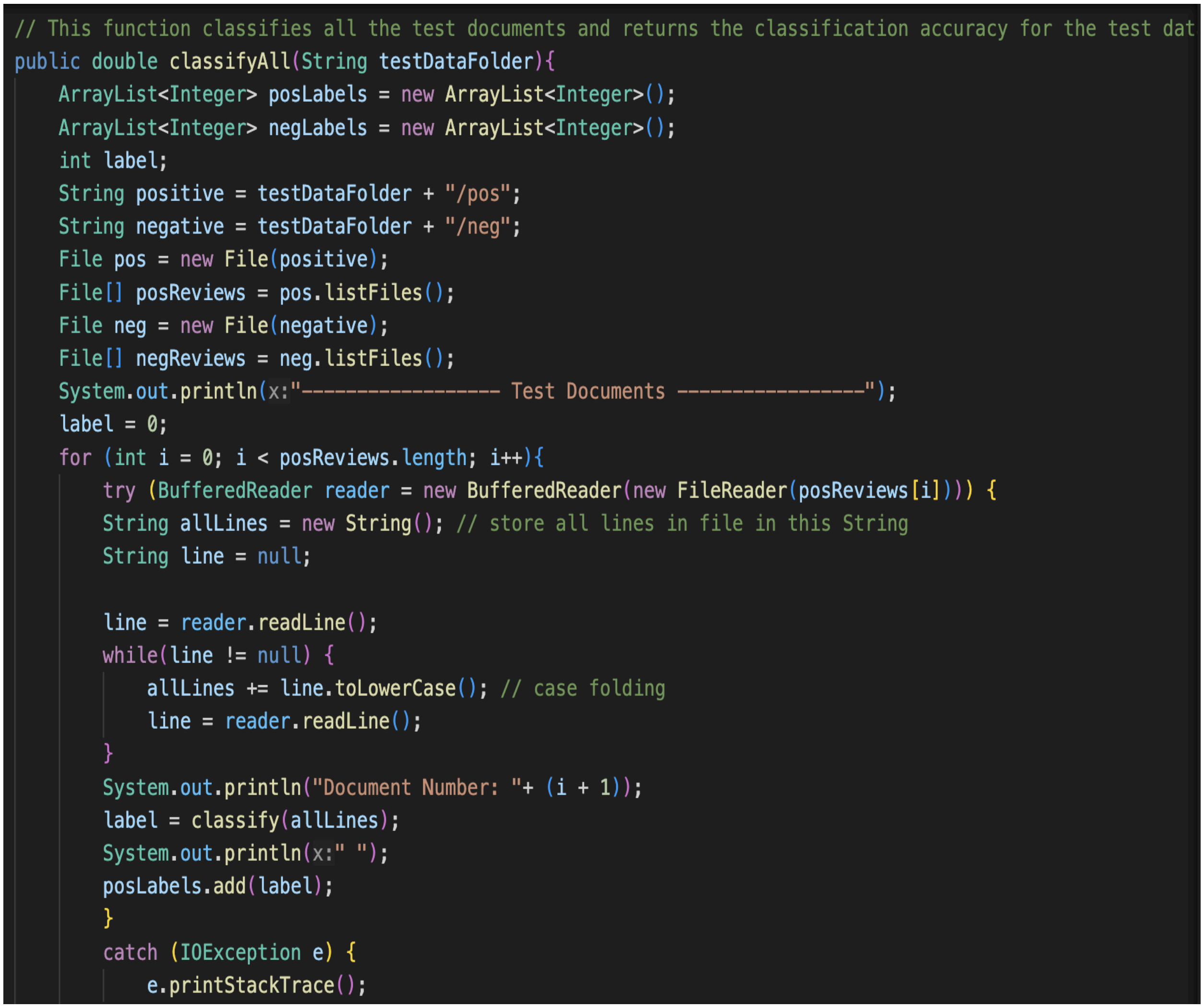
```java
// This function is basically used to initialize the integer array of class labels for the training documen
// I have done the rest of the initializations in the constructor
public void preprocess(String trainDataFolder) {
    ArrayList<Integer> classLabels = new ArrayList<Integer>();
    String positive = trainDataFolder + "/pos";
    String negative = trainDataFolder + "/neg";
    File pos = new File(positive);
    File[] posReviews = pos.listFiles();
    File neg = new File(negative);
    File[] negReviews = neg.listFiles();
    for (int i = 0; i < posReviews.length; i++){
        try (BufferedReader reader = new BufferedReader(new FileReader(posReviews[i]))) {
        String allLines = new String(); // store all lines in file in this String
        String line = null;

        line = reader.readLine();
        while(line != null) {
            allLines += line.toLowerCase(); // case folding
            line = reader.readLine();
        }
        trainDocs.add(allLines);
        classLabels.add(e:0);
    }
    catch (IOException e) {
        e.printStackTrace();
    }
    }
```

Task 2 (20 points): Classifying individual testing documents



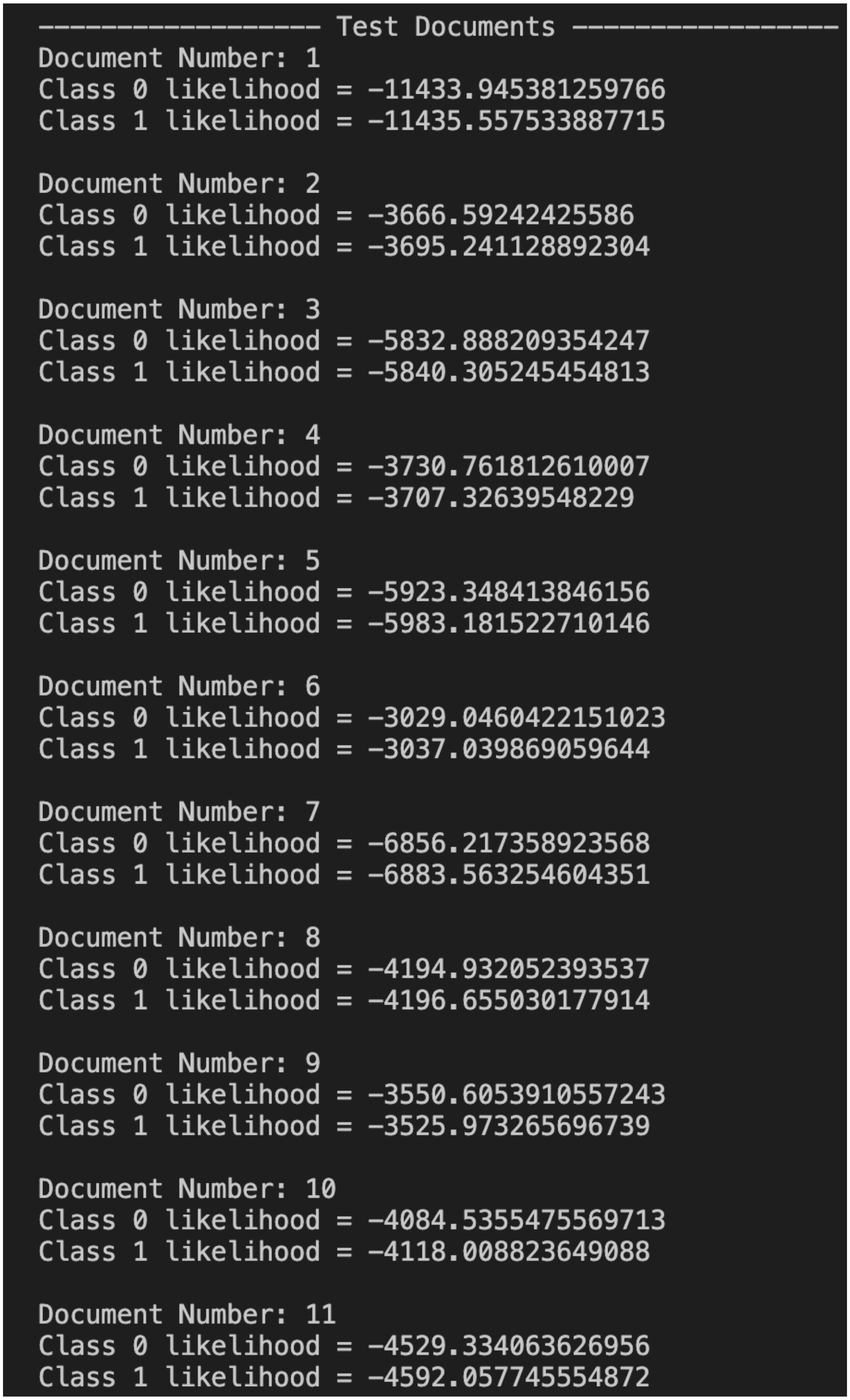
```java
// This function is used to classify each test document and returns a class value on the basis of likelihoo
public int classify(String testDoc){
    int label = 0;
    int vSize = vocabulary.size();
    double[] score = new double[numClasses]; // class likelihood for each class

    for(int i = 0; i < score.length; i++) {
        // use log to avoid precision problems
        score[i] = Math.log(classCounts[i] * 1.0 / trainDocs.size()); // prior probability of class
    }

    String[] tokens = testDoc.split(regex:"[ .,?!:;$#%*+/()\"\\-&]+");

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

        for(String token: tokens) {
            // use log/addition to avoid precision problems
            if(condProb[i].containsKey(token))
                score[i] += Math.log(condProb[i].get(token)); // term's class conditional probability
            else
                score[i] += Math.log(1.0 / (classTokenCounts[i] + vSize)); // previously unknown term, comp
        }
    }

    double maxScore = score[0];

    // find the largest class likelihood and save its label to return as the class value
    for(int i = 0; i < score.length; i++) {
        System.out.println("Class " + i + " likelihood = " + score[i]);
        if(score[i] > maxScore) {
```

Task 3 (40 points): Classify all the test documents in the test folder and report the classification accuracy

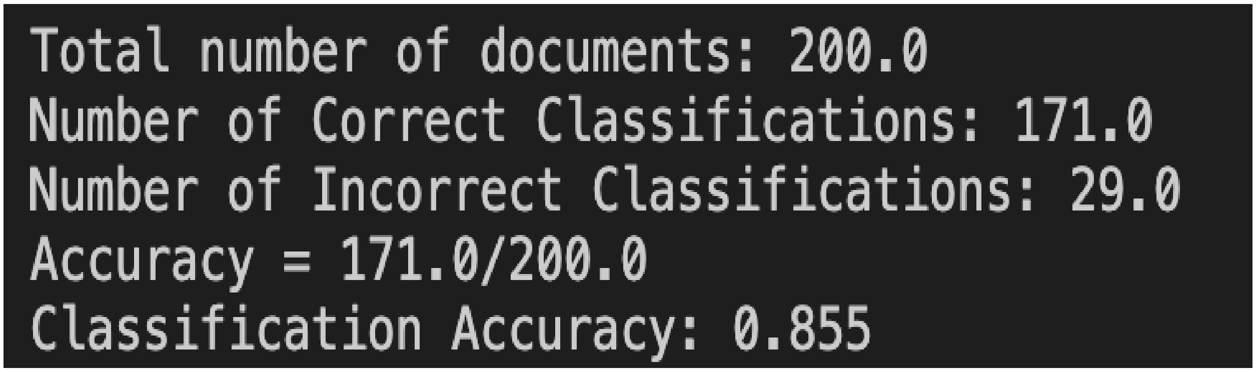


```java
// This function classifies all the test documents and returns the classification accuracy for the test dat
public double classifyAll(String testDataFolder){
    ArrayList<Integer> posLabels = new ArrayList<Integer>();
    ArrayList<Integer> negLabels = new ArrayList<Integer>();
    int label;
    String positive = testDataFolder + "/pos";
    String negative = testDataFolder + "/neg";
    File pos = new File(positive);
    File[] posReviews = pos.listFiles();
    File neg = new File(negative);
    File[] negReviews = neg.listFiles();
    System.out.println(x:"------------------ Test Documents -----------------");
    label = 0;
    for (int i = 0; i < posReviews.length; i++){
        try (BufferedReader reader = new BufferedReader(new FileReader(posReviews[i]))) {
        String allLines = new String(); // store all lines in file in this String
        String line = null;

        line = reader.readLine();
        while(line != null) {
            allLines += line.toLowerCase(); // case folding
            line = reader.readLine();
        }
        System.out.println("Document Number: "+ (i + 1));
        label = classify(allLines);
        System.out.println(x:" ");
        posLabels.add(label);
        }
        catch (IOException e) {
            e.printStackTrace();
```

Final Results:



```
------------------ Test Documents -----------------
Document Number: 1
Class 0 likelihood = -11433.945381259766
Class 1 likelihood = -11435.557533887715

Document Number: 2
Class 0 likelihood = -3666.59242425586
Class 1 likelihood = -3695.241128892304

Document Number: 3
Class 0 likelihood = -5832.888209354247
Class 1 likelihood = -5840.305245454813

Document Number: 4
Class 0 likelihood = -3730.761812610007
Class 1 likelihood = -3707.32639548229

Document Number: 5
Class 0 likelihood = -5923.348413846156
Class 1 likelihood = -5983.181522710146

Document Number: 6
Class 0 likelihood = -3029.0460422151023
Class 1 likelihood = -3037.039869059644

Document Number: 7
Class 0 likelihood = -6856.217358923568
Class 1 likelihood = -6883.563254604351

Document Number: 8
Class 0 likelihood = -4194.932052393537
Class 1 likelihood = -4196.655030177914

Document Number: 9
Class 0 likelihood = -3550.6053910557243
Class 1 likelihood = -3525.973265696739

Document Number: 10
Class 0 likelihood = -4084.5355475569713
Class 1 likelihood = -4118.008823649088

Document Number: 11
Class 0 likelihood = -4529.334063626956
Class 1 likelihood = -4592.057745554872
```



```
Total number of documents: 200.0
Number of Correct Classifications: 171.0
Number of Incorrect Classifications: 29.0
Accuracy = 171.0/200.0
Classification Accuracy: 0.855
```