Abigail Smith

ARS190011

CS 4375.004

Dr. Mazidi

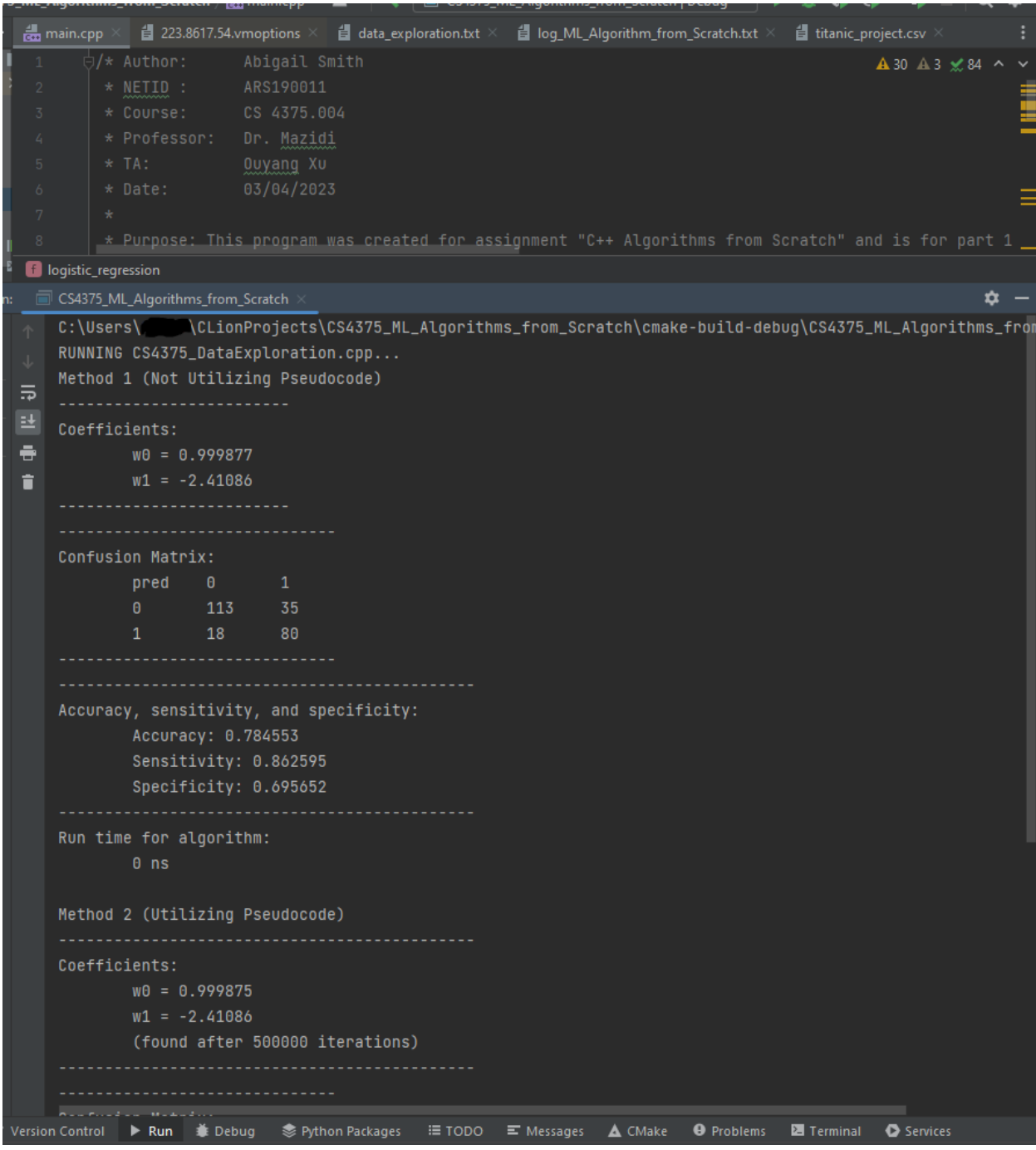
TA: Ouyang Xu

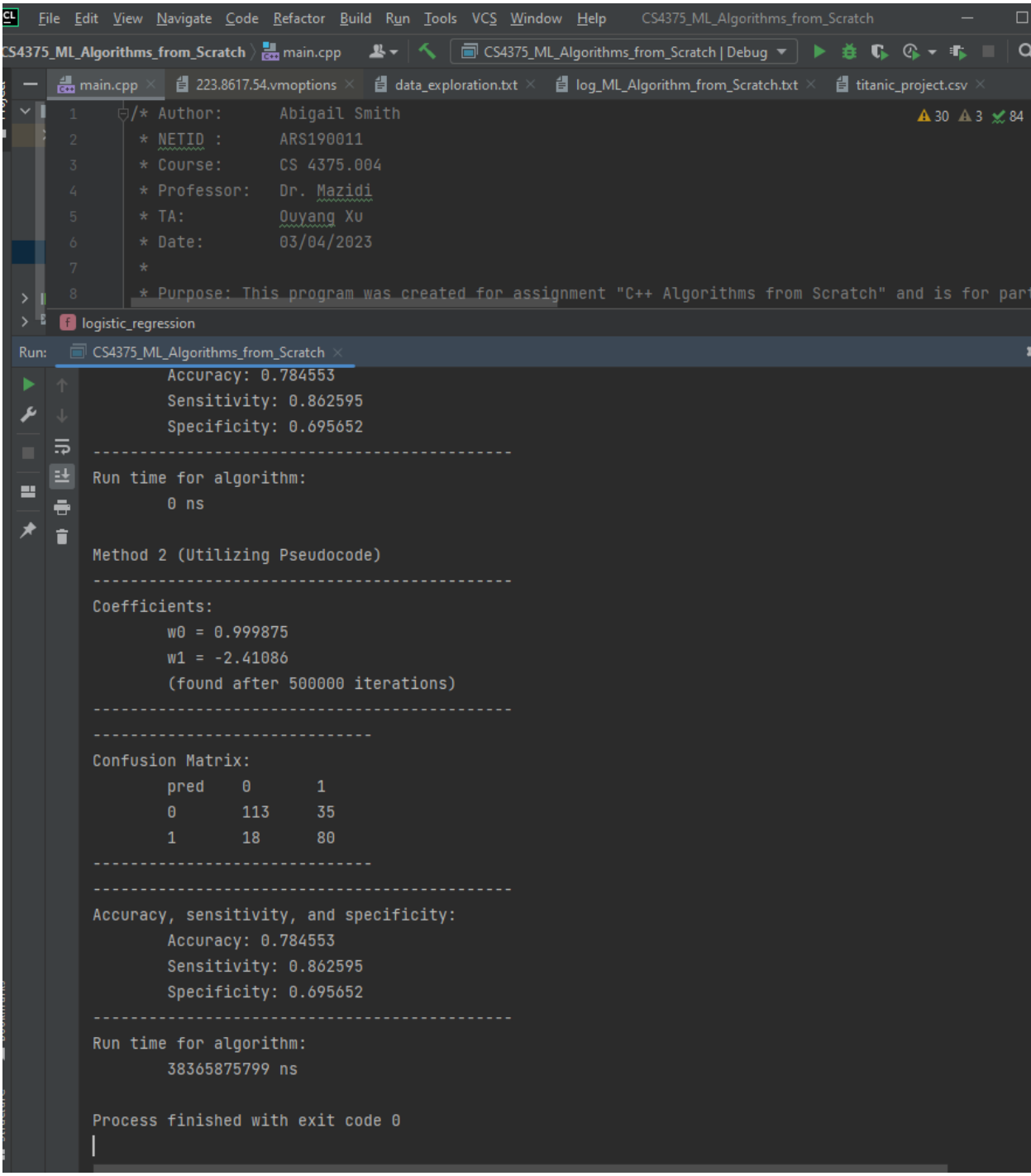
03/04/2023

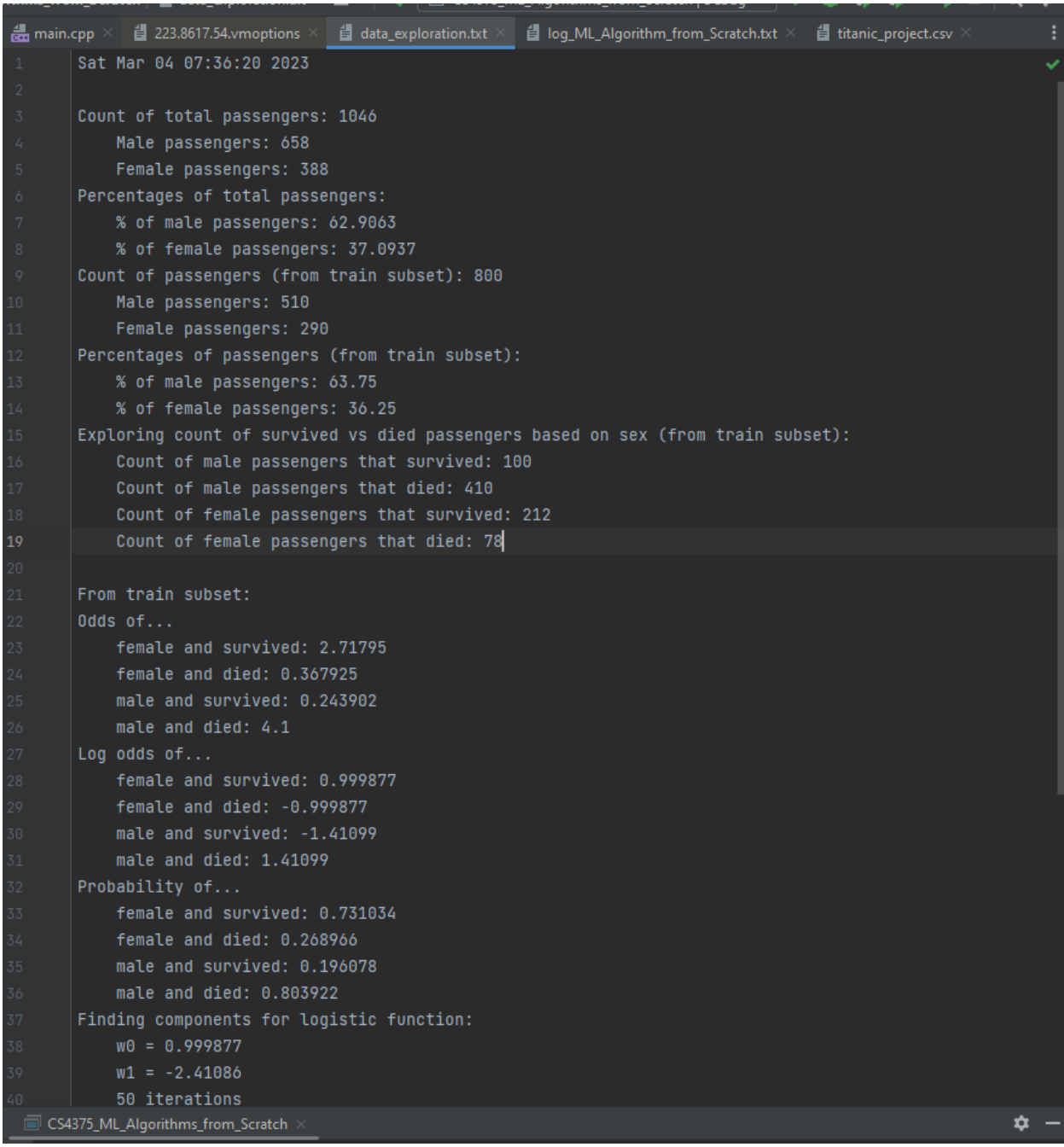
ML Algorithms from Scratch

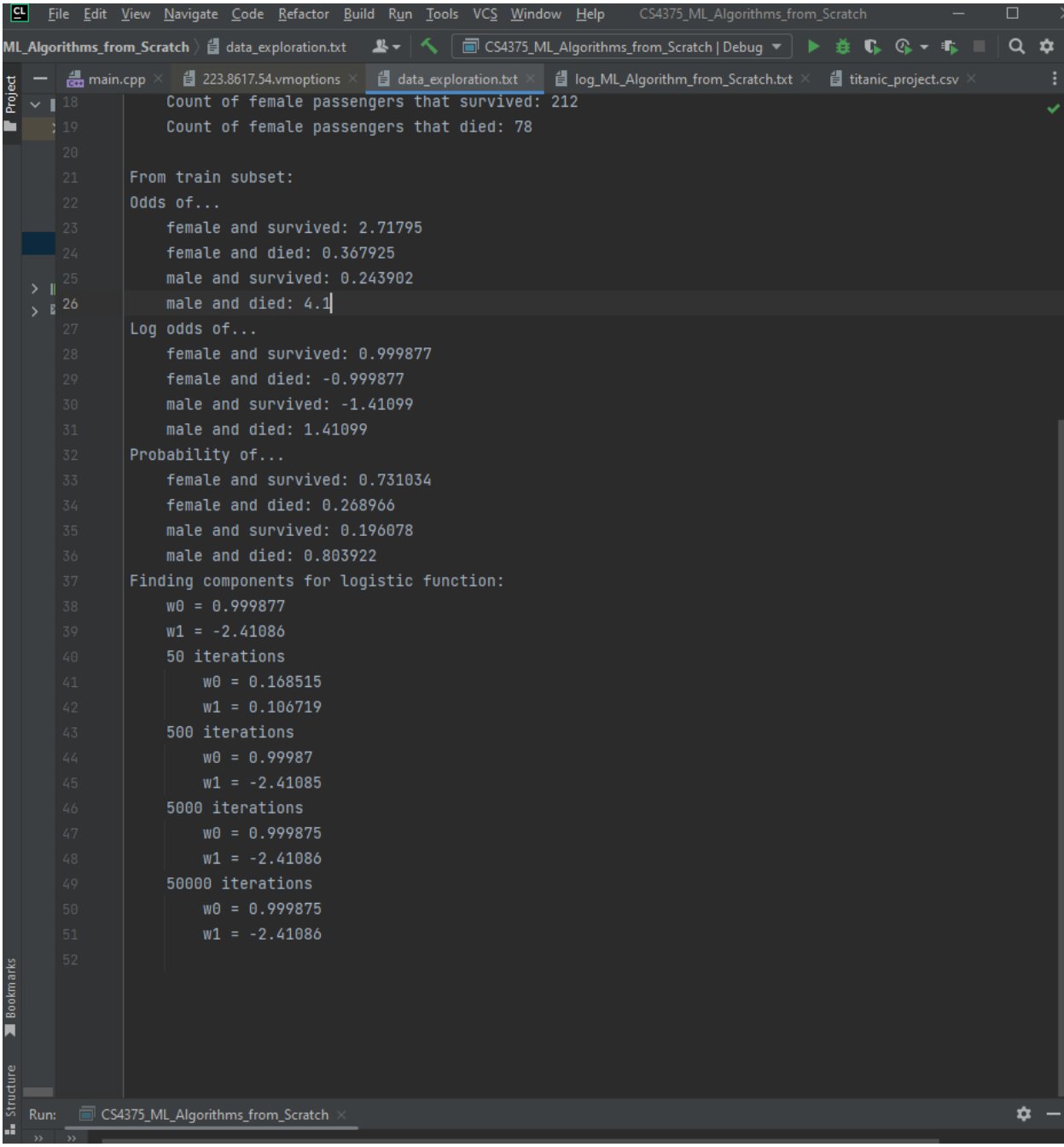
A. Copy/paste runs of your code showing the output (coefficients and metrics), and run times

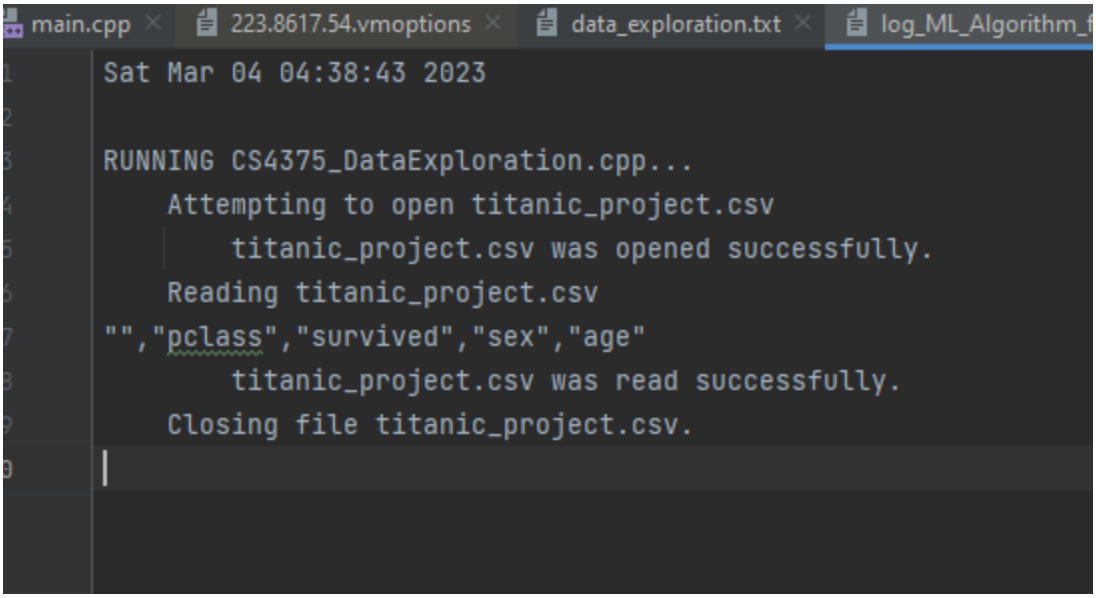
   I. Program 1 (Logistic Regression)


main.cpp
223.8617.54.vmoptions
data_exploration.txt
log_ML_Algorithm_from_Scratch.txt
titanic_project.csv
/* Author: Abigail Smith
* NETID : ARS190011
* Course: CS 4375.004
* Professor: Dr. Mazidi
* TA: Ouyang Xu
* Date: 03/04/2023
*
* Purpose: This program was created for assignment "C++ Algorithms from Scratch" and is for part 1
logistic_regression
CS4375_ML_Algorithms_from_Scratch
C:\Users\[redacted]\CLionProjects\CS4375_ML_Algorithms_from_Scratch\cmake-build-debug\CS4375_ML_Algorithms_fro
RUNNING CS4375_DataExploration.cpp...
Method 1 (Not Utilizing Pseudocode)
Coefficients:
w0 = 0.999877
w1 = -2.41086
Confusion Matrix:
pred 0 1
0 113 35
1 18 80
Accuracy, sensitivity, and specificity:
Accuracy: 0.784553
Sensitivity: 0.862595
Specificity: 0.695652
Run time for algorithm:
0 ns
Method 2 (Utilizing Pseudocode)
Coefficients:
w0 = 0.999875
w1 = -2.41086
(found after 500000 iterations)
Version Control
Run
Debug
Python Packages
TODO
Messages
CMake
Problems
Terminal
Services

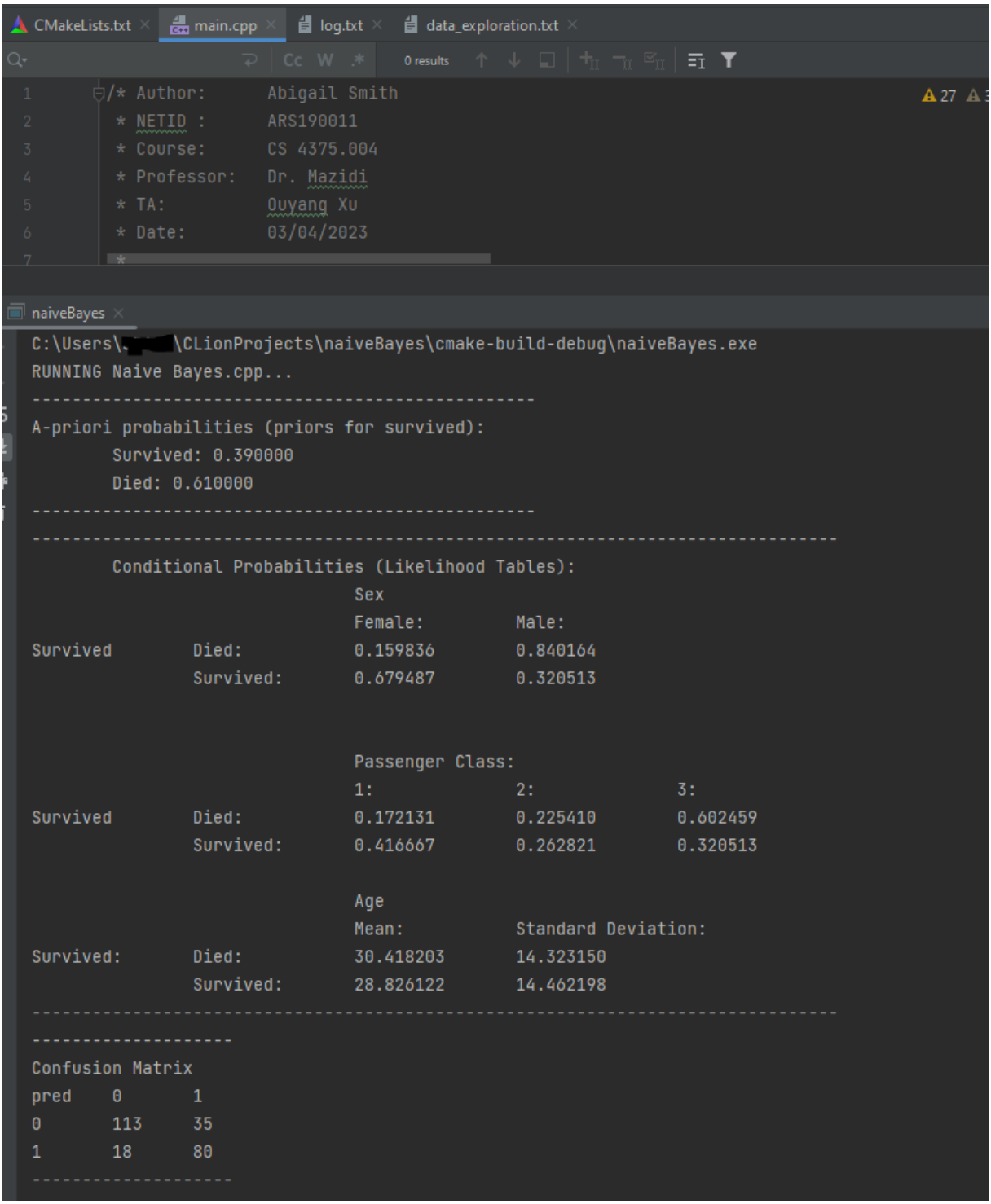

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
CS4375_ML_Algorithms_from_Scratch
CS4375_ML_Algorithms_from_Scratch | Debug
main.cpp
223.8617.54.vmoptions
data_exploration.txt
log_ML_Algorithm_from_Scratch.txt
titanic_project.csv
/* Author: Abigail Smith
* NETID : ARS190011
* Course: CS 4375.004
* Professor: Dr. Mazidi
* TA: Ouyang Xu
* Date: 03/04/2023
*
* Purpose: This program was created for assignment "C++ Algorithms from Scratch" and is for part
logistic_regression
Run:
CS4375_ML_Algorithms_from_Scratch
Accuracy: 0.784553
Sensitivity: 0.862595
Specificity: 0.695652
Run time for algorithm:
0 ns
Method 2 (Utilizing Pseudocode)
Coefficients:
w0 = 0.999875
w1 = -2.41086
(found after 500000 iterations)
Confusion Matrix:
pred 0 1
0 113 35
1 18 80
Accuracy, sensitivity, and specificity:
Accuracy: 0.784553
Sensitivity: 0.862595
Specificity: 0.695652
Run time for algorithm:
38365875799 ns
Process finished with exit code 0


main.cpp
223.8617.54.vmoptions
data_exploration.txt
log_ML_Algorithm_from_Scratch.txt
titanic_project.csv
Sat Mar 04 07:36:20 2023
Count of total passengers: 1046
Male passengers: 658
Female passengers: 388
Percentages of total passengers:
% of male passengers: 62.9063
% of female passengers: 37.0937
Count of passengers (from train subset): 800
Male passengers: 510
Female passengers: 290
Percentages of passengers (from train subset):
% of male passengers: 63.75
% of female passengers: 36.25
Exploring count of survived vs died passengers based on sex (from train subset):
Count of male passengers that survived: 100
Count of male passengers that died: 410
Count of female passengers that survived: 212
Count of female passengers that died: 78
From train subset:
Odds of...
female and survived: 2.71795
female and died: 0.367925
male and survived: 0.243902
male and died: 4.1
Log odds of...
female and survived: 0.999877
female and died: -0.999877
male and survived: -1.41099
male and died: 1.41099
Probability of...
female and survived: 0.731034
female and died: 0.268966
male and survived: 0.196078
male and died: 0.803922
Finding components for logistic function:
w0 = 0.999877
w1 = -2.41086
50 iterations
CS4375_ML_Algorithms_from_Scratch


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
CS4375_ML_Algorithms_from_Scratch
ML_Algorithms_from_Scratch
data_exploration.txt
CS4375_ML_Algorithms_from_Scratch | Debug
main.cpp
223.8617.54.vmoptions
data_exploration.txt
log_ML_Algorithm_from_Scratch.txt
titanic_project.csv
Count of female passengers that survived: 212
Count of female passengers that died: 78
From train subset:
Odds of...
female and survived: 2.71795
female and died: 0.367925
male and survived: 0.243902
male and died: 4.1
Log odds of...
female and survived: 0.999877
female and died: -0.999877
male and survived: -1.41099
male and died: 1.41099
Probability of...
female and survived: 0.731034
female and died: 0.268966
male and survived: 0.196078
male and died: 0.803922
Finding components for logistic function:
w0 = 0.999877
w1 = -2.41086
50 iterations
w0 = 0.168515
w1 = 0.106719
500 iterations
w0 = 0.99987
w1 = -2.41085
5000 iterations
w0 = 0.999875
w1 = -2.41086
50000 iterations
w0 = 0.999875
w1 = -2.41086
Run:
CS4375_ML_Algorithms_from_Scratch


main.cpp
223.8617.54.vmoptions
data_exploration.txt
log_ML_Algorithm_f
Sat Mar 04 04:38:43 2023
RUNNING CS4375_DataExploration.cpp...
Attempting to open titanic_project.csv
titanic_project.csv was opened successfully.
Reading titanic_project.csv
"","pclass","survived","sex","age"
titanic_project.csv was read successfully.
Closing file titanic_project.csv.

II. Program 2 (Naive Bayes)



CMakeLists.txt | main.cpp | log.txt | data_exploration.txt

0 results

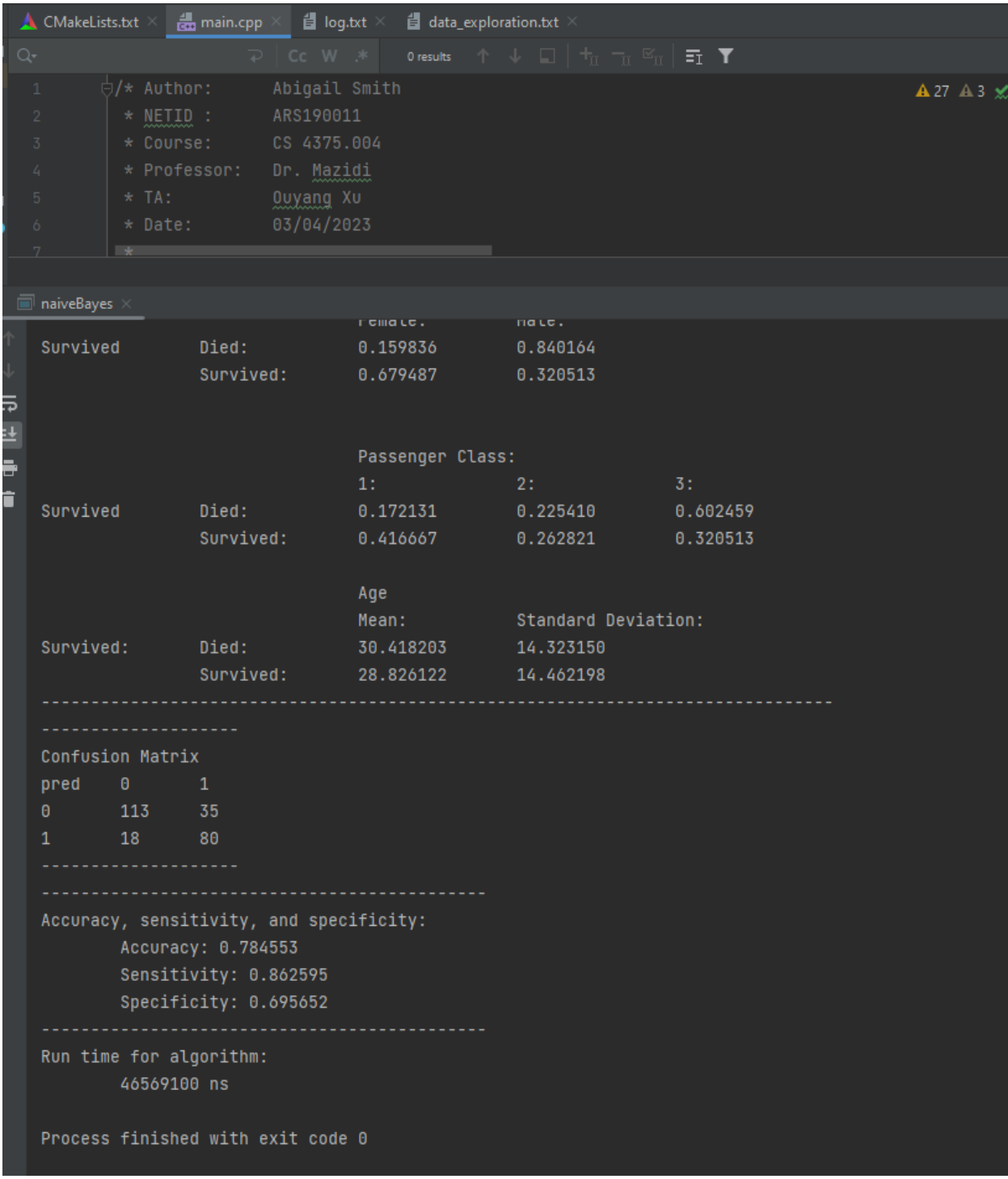
```
/* Author:      Abigail Smith
 * NETID :      ARS190011
 * Course:      CS 4375.004
 * Professor:   Dr. Mazidi
 * TA:          Ouyang Xu
 * Date:        03/04/2023
 *
```

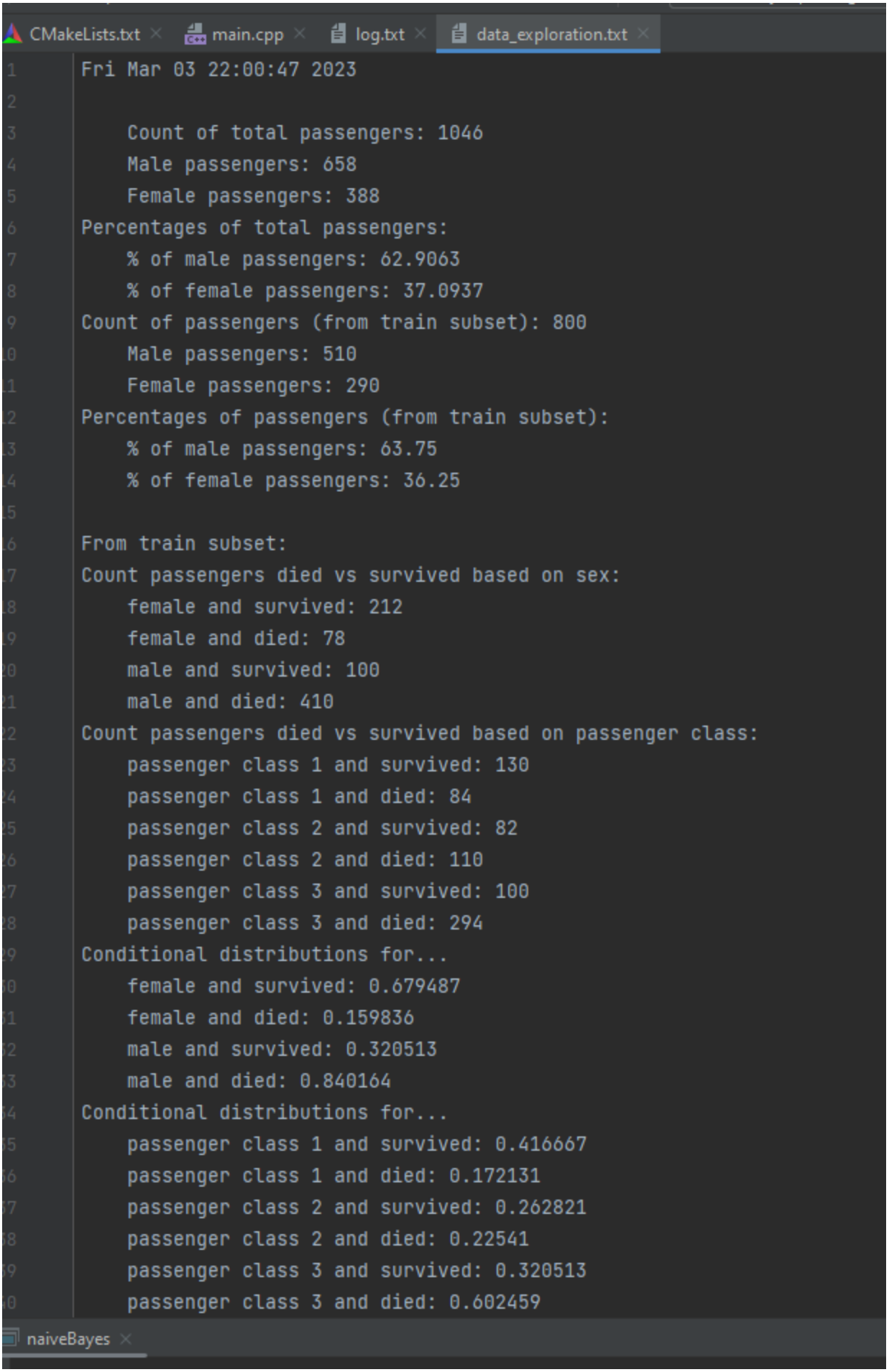
naiveBayes

```
C:\Users\[redacted]\CLionProjects\naiveBayes\cmake-build-debug\naiveBayes.exe
RUNNING Naive Bayes.cpp...
-------------------------------------------------
A-priori probabilities (priors for survived):
        Survived: 0.390000
        Died: 0.610000
-------------------------------------------------
--------------------------------------------------------------------------------
        Conditional Probabilities (Likelihood Tables):
                                Sex
                                Female:         Male:
Survived        Died:           0.159836        0.840164
                Survived:       0.679487        0.320513


                                Passenger Class:
                                1:              2:              3:
Survived        Died:           0.172131        0.225410        0.602459
                Survived:       0.416667        0.262821        0.320513

                                Age
                                Mean:           Standard Deviation:
Survived:       Died:           30.418203       14.323150
                Survived:       28.826122       14.462198
--------------------------------------------------------------------------------
-------------------
Confusion Matrix
pred    0       1
0       113     35
1       18      80
-------------------
```


CMakeLists.txt
main.cpp
log.txt
data_exploration.txt
0 results
/* Author: Abigail Smith
* NETID : ARS190011
* Course: CS 4375.004
* Professor: Dr. Mazidi
* TA: Ouyang Xu
* Date: 03/04/2023
naiveBayes
Survived Died: 0.159836 0.840164
Survived: 0.679487 0.320513
Passenger Class:
1: 2: 3:
Survived Died: 0.172131 0.225410 0.602459
Survived: 0.416667 0.262821 0.320513
Age
Mean: Standard Deviation:
Survived: Died: 30.418203 14.323150
Survived: 30.826122 14.462198
Confusion Matrix
pred 0 1
0 113 35
1 18 80
Accuracy, sensitivity, and specificity:
Accuracy: 0.784553
Sensitivity: 0.862595
Specificity: 0.695652
Run time for algorithm:
46569100 ns
Process finished with exit code 0


CMakeLists.txt
main.cpp
log.txt
data_exploration.txt
Fri Mar 03 22:00:47 2023
Count of total passengers: 1046
Male passengers: 658
Female passengers: 388
Percentages of total passengers:
% of male passengers: 62.9063
% of female passengers: 37.0937
Count of passengers (from train subset): 800
Male passengers: 510
Female passengers: 290
Percentages of passengers (from train subset):
% of male passengers: 63.75
% of female passengers: 36.25
From train subset:
Count passengers died vs survived based on sex:
female and survived: 212
female and died: 78
male and survived: 100
male and died: 410
Count passengers died vs survived based on passenger class:
passenger class 1 and survived: 130
passenger class 1 and died: 84
passenger class 2 and survived: 82
passenger class 2 and died: 110
passenger class 3 and survived: 100
passenger class 3 and died: 294
Conditional distributions for...
female and survived: 0.679487
female and died: 0.159836
male and survived: 0.320513
male and died: 0.840164
Conditional distributions for...
passenger class 1 and survived: 0.416667
passenger class 1 and died: 0.172131
passenger class 2 and survived: 0.262821
passenger class 2 and died: 0.22541
passenger class 3 and survived: 0.320513
passenger class 3 and died: 0.602459
naiveBayes

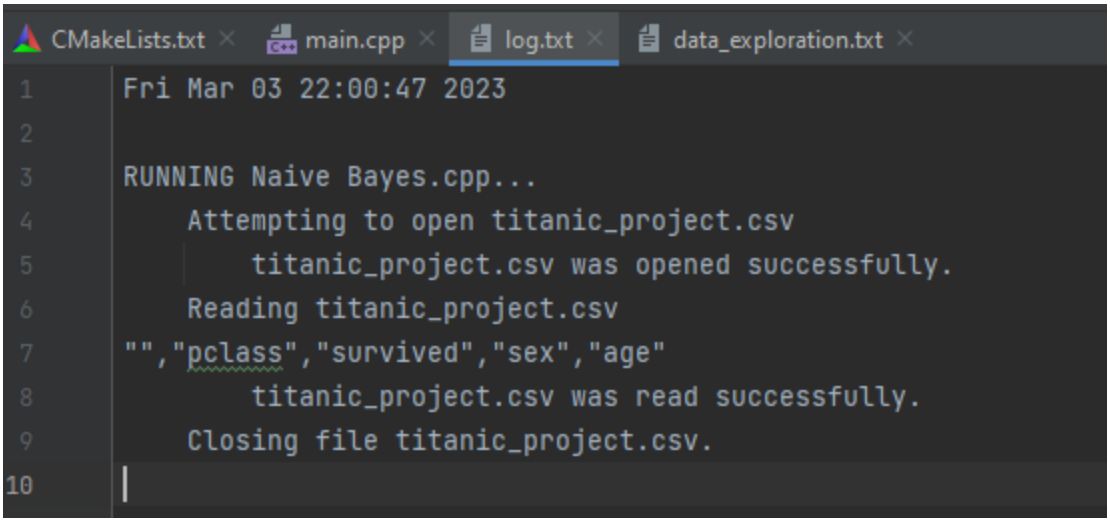
CMakeLists.txt | main.cpp | log.txt | data_exploration.txt

```
Percentages of passengers (from train subset):
    % of male passengers: 63.75
    % of female passengers: 36.25

From train subset:
Count passengers died vs survived based on sex:
    female and survived: 212
    female and died: 78
    male and survived: 100
    male and died: 410
Count passengers died vs survived based on passenger class:
    passenger class 1 and survived: 130
    passenger class 1 and died: 84
    passenger class 2 and survived: 82
    passenger class 2 and died: 110
    passenger class 3 and survived: 100
    passenger class 3 and died: 294
Conditional distributions for...
    female and survived: 0.679487
    female and died: 0.159836
    male and survived: 0.320513
    male and died: 0.840164
Conditional distributions for...
    passenger class 1 and survived: 0.416667
    passenger class 1 and died: 0.172131
    passenger class 2 and survived: 0.262821
    passenger class 2 and died: 0.22541
    passenger class 3 and survived: 0.320513
    passenger class 3 and died: 0.602459
Conditional distributions for...
    Age (Survived) (Mean): 28.8261
    Age (Survived) (Standard Deviation): 14.4622
    Age (Died) (Mean): 30.4182
    Age (Died) (Standard Deviation): 14.3231

```

naiveBayes


CMakeLists.txt
main.cpp
log.txt
data_exploration.txt
Fri Mar 03 22:00:47 2023
RUNNING Naive Bayes.cpp...
Attempting to open titanic_project.csv
titanic_project.csv was opened successfully.
Reading titanic_project.csv
"","pclass","survived","sex","age"
titanic_project.csv was read successfully.
Closing file titanic_project.csv.

B. Analyze the results of your algorithms on the Titanic data.

In the logistic regression program, the Titanic data is undergoing classification in order to identify if a passenger has survived or died based on their sex. I would like to take a moment to explain my outputs for logistic regression. When I first worked on this assignment, I found a way to find logistic regression without using the psuedocode from the textbook, and after finding this, I then used the psuedocode logic (that is why there is a “Method 1” and “Method 2” section). I kept both algorithms in program one since I thought they both were interesting.

Finding logistic regression required method 1 to find the odds, log odds, probability of survival (based on sex), and coefficients w0 and w1 in order to create the logistic regression formula p(x) = 1 / (1 + e^-(w0 + w1x). In contrast, method 2 utilized gradient descent to find optimal coefficients through a sigmoid function. Unlike linear regression, logistic regression outputs probabilities in the range [0, 1] where the cutoff point between classifications is, usually, 0.5 (the inflection point). In order to categorize the test subset, the logistic regression formula utilized the current observation’s sex as x in the logistic regression formula, and as described before, the cutoff point was used to classify passengers as survived versus died. From this classification, a confusion matrix, accuracy, sensitivity, and specificity were calculated.

In order to find the coefficients w0 and w1, method one had to find the probabilities for survival based on sex. Probability represents the probability that something will happen and is found by dividing the odds of something happening by the odds of something happening plus one (i.e. probability = (odds) / (odds + 1)). In order to find probability, the odds of survival based on sex had to be calculated by finding the ratios for survival based on sex. From these components, w0 and w1 were able to be found and used in the logistic regression formula and, thus, able to produce the results of program 1.

In contrast, method 2 simply utilized gradient descent to find the optimal coefficient by creating a sigmoid function. Weights w0 and w1 were gradually adjusted as more data was explored, bringing the calculated weights closer and closer to the weights produced by the glm() function in R with each additional iteration. In order to find these weights, both the sigmoid function and error was used in conjunction with a learning rate to gradually adjust w0 and w1 each iteration.

From the logistic regression results, the coefficients w0 and w1 represent the intercept and slope (respectively) -- or better said -- the coefficient quantifies the difference in the log odds for a 1-unit change in x. In regards to the output, a w0 = 0.999877 and a w1 = -2.41086 means that, for every one unit change in sex, the log odds of survival increased by -2.41086. From quick data exploration, a w1 = -2.41086 is reasonable given there are only two sex values (0 representing female and 1 representing male), and as seen in data exploration, there were significantly more men that died than women in the training subset (410 men died and 78 women died).

The output confusion matrix represents the predictions made from the logistic regression and whether they were correctly identified or not. The logistic regression program was able to correctly identify 113 deaths and 80 survivals but incorrectly identified 35 deaths and 18 survivals. By having an accuracy of nearly 78%, this confusion matrix is arguable proof of a successful classification algorithm. This is only fortified by a sensitivity of nearly 86%, or an 86% true positive rate, and a specificity of nearly 69%, or a 69% true negative rate (meaning the model was more likely to correctly identify a death than survival, which makes sense given 488 people had died while only 312 had survived).

In the Naive Bayes program, the Titanic data is classified in order to identify if a passenger has survived based on age, sex, and passenger class. Finding Naive Bayes required the second program to find the posterior, probabilities, conditional probabilities, and likelihood. In order to categorize the test subset, the Naive Bayes formula was utilized to categorize each test observation into survived or died.

The first output result for the Naive Bayes program is the priors for survival. As seen in data exploration, there are more instances where a person has died than survived in the test subset. Because of this, the prior for survived = 0.39 and the prior for died = 0.61 is understandable.

The next output to review are the conditional probabilities (or likelihoods). The conditional probabilities represent whether a passenger survived or died based on sex, age, or passenger class. The conditional probabilities of survived based on sex and survived based on passenger class are understandable since, as seen in the data exploration, female passengers were less likely to die, male passengers were more likely to die, and passenger class one was most likely to survive followed by passenger class 3. The conditional probabilities for age are based on mean and standard deviation since age, unlike sex and passenger class, is continuous.

Like in logistic regression, the Naive Bayes program produced a confusion matrix. The Naive Bayes program was able to correctly identify 113 deaths and 80 survivals but incorrectly identified 35 deaths and 18 survivals. By having an accuracy of nearly 78%, this confusion matrix is arguable proof of a successful classification algorithm. This is only fortified by a sensitivity of nearly 86%, or an 86% true positive rate, and a specificity of nearly 69%, or a 69% true negative rate (meaning the model was more likely to correctly identify a death than survival, which makes sense given 488 people had died while only 312 had survived).

One important note for both the logistic regression program and the Naive Bayes program is that the provided test data was relatively small, meaning a stronger model could be made with a larger data set.

C. write two paragraphs comparing and contrasting generative classifiers versus discriminative classifiers.

In terms of classification, generative classifiers and discriminative classifiers represent two different approaches to classification. Generative classifiers work to use the joint probability distribution to understand how the data is distributed in order to perform classification, as seen with Naive Bayes. In contrast, discriminative classifiers aim to find boundaries to separate classes, as seen with logistic regression. In order to perform classification, generative classification must make the assumption that all features are conditionally independent while, in contrast, discriminative classification does not make any assumptions. Generative classification makes assumptions due to its functionality -- unlike discriminative classification, generative classification, as in the name, generates a probability based on the distribution of the dataset to perform classification. Discriminative classification does not make assumptions since, by its functionality, it utilizes the data present to create a boundary for classification.

By their structure, generative classification and discriminative classification feature unique limitations and applications. By its nature, discriminative classification is better suited to handle outliers than generative classification (though it is important to note that discriminative classification is still susceptible to misclassification despite this advantage). Additionally, it is important to note that in terms of applications, discriminative classification is more applicable to supervised learning while generative classification is more useful for unsupervised learning. This distinction comes from how these classification models classify data.

Sources:

Goyal, Chirag. “2023's Best Guide to Descriptive & Generative Machine Learning Models.” *Analytics Vidhya*, 19 July 2021, https://www.analyticsvidhya.com/blog/2021/07/deep-understanding-of-discriminative-and-generative-models-in-machine-learning/.

D. Google this phrase: reproducible research in machine learning.

In machine learning, reproducibility is a critical foundation for credible, meaningful research. As in other scientific fields, reproducibility acts as a defining feature for credibility, and in machine learning refers to having the ability to run an algorithm various times "on certain datasets and obtain the same (or similar) results" ("The Importance of Reproducibility"). Reproducibility provides a guarantee that the results of a machine learning model accurately evaluate and apply the utilized data set while simultaneously reinforcing the results of the machine learning model. Reproducible research in machine learning means that from one workflow to the next, a machine learning application will arrive at the same conclusions.

Often, failure to create reproducible research in machine learning derives from a variety of factors. An individual or team may fail to replicate research in machine learning due to "'missing raw or original data, a lack of tidied up version of the data, no source code available, or lacking the software to run the experiment'" as well as a "'lack of documentation and deprecated dependencies'" (Fonseca Cacho). Each of these factors presents a unique challenge that poses a threat to the reproducibility of a machine learning model. Without countering each of these challenges, the chances of successfully reproducing the results of machine learning research decreases.

Reproducibility acts as a critical foundation for creating meaningful machine learning research. At its core, reproducibility helps to "reduce errors and ambiguity" and "ensures data consistency", two features critical not only to ensure the accuracy of a machine learning model, but also to fortify credibility ("The Importance of Reproducibility"). Without reproducibility, a machine learning model's results would require constant scrutiny brought on by an inability to validate the model's outputs. Reproducibility directly supports the importance of a machine

learning application's findings by giving these findings a backing not only to the individual or team that originally created the machine learning model, but to those looking to utilize its findings.

Various ways exist to ensure reproducibility in a machine learning model. One way is to ensure the replication utilizes the same environment as the original machine learning model since "the hardware or dependencies surrounding the source code" may "become outdated or deprecated making it very complicated to run old code" (Fonseca Cacho). These features, though seemingly unimportant, work to create a virtual lab that provides the same structure and validity as a physical lab to the physical sciences. Without implementing the correct tools, an attempt to replicate the findings of a machine learning model may fail since the correct steps were not taken to completely replicate the model.

Beyond using the same hardware or dependencies utilized by the source code, documentation is another critical step in ensuring reproducibility. From the start of a machine learning endeavor, documentation should be created to "explain why certain choices were made, as well as a range of important details needed to successfully execute the project" ("The Importance of Reproducibility"). Documentation ensures that those looking to replicate a machine learning model can make sure they have all the resources required to successfully replicate the findings of the machine learning application. Without documentation explaining why and how a machine learning model works, the chances of successfully replicating the model decrease.

Overall, reproducibility acts as the foundation for meaningful and credible machine learning research. By understanding what exactly reproducibility means in machine learning

research, understanding its important, and how to implement it within projects, the validity and credibility of a machine learning project will drastically increase.

Sources:

Fonseca Cacho, Jorge Ramón, and Kazem Taghva. “The State of Reproducible Research in Computer Science.” *Advances in Intelligent Systems and Computing*, vol. 1134, 2020, pp. 519–524., https://doi.org/10.1007/978-3-030-43020-7_68.

“The Importance of Reproducibility in Machine Learning Applications.” *DecisivEdge*, https://www.decisivedge.com/blog/the-importance-of-reproducibility-in-machine-learning-applications/#:~:text=Reproducibility%20with%20respect%20to%20machine,reporting%2C%20data%20analysis%20and%20interpretation.