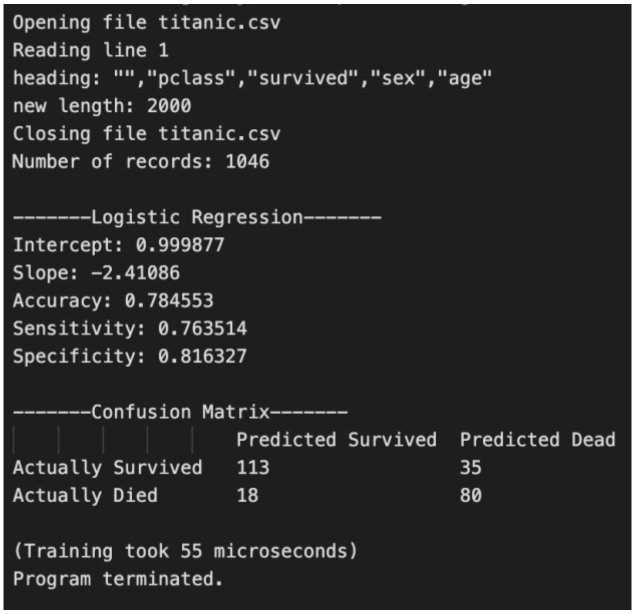
Code Run Outputs:

Naïve Bayes ----------------------------------->

Logistic Regression

|
|
|
|
|
|
|
v

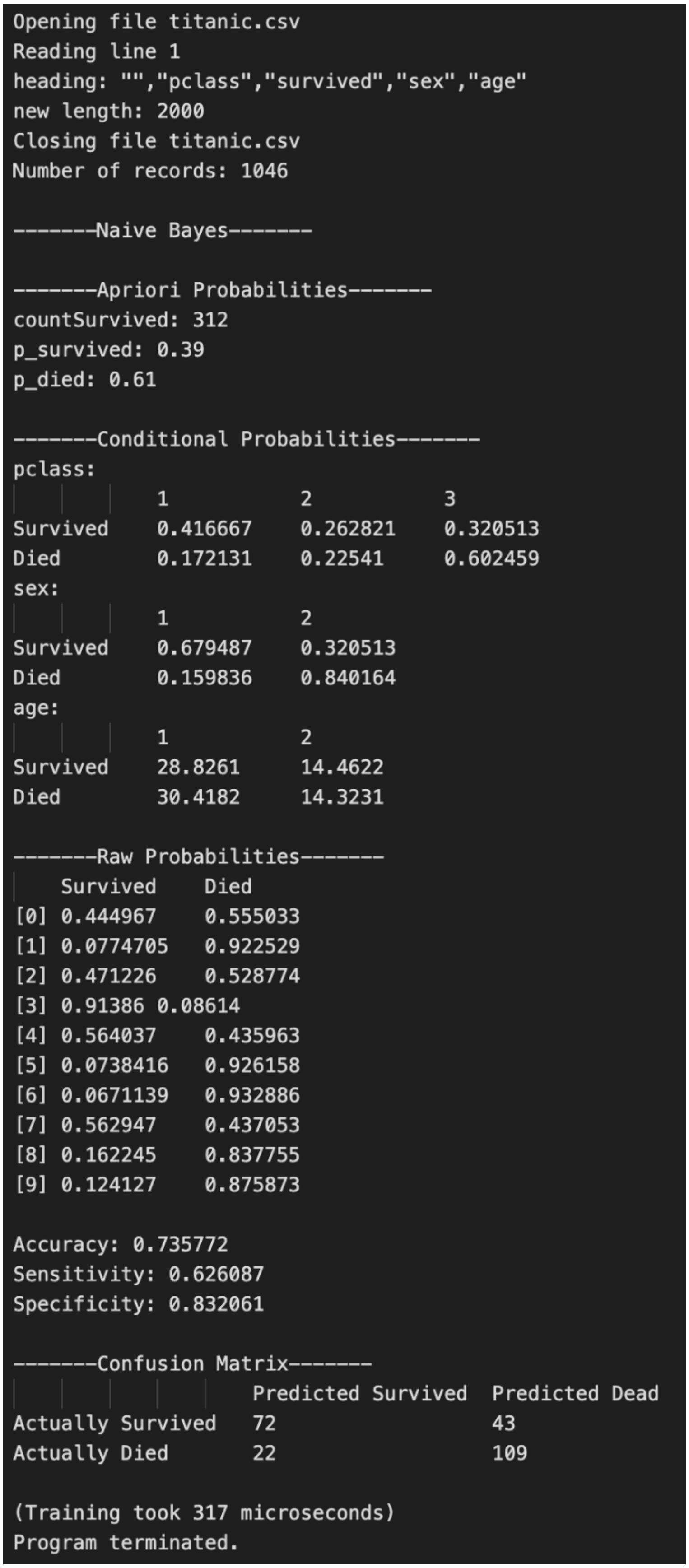


```
Opening file titanic.csv
Reading line 1
heading: "","pclass","survived","sex","age"
new length: 2000
Closing file titanic.csv
Number of records: 1046

-------Logistic Regression-------
Intercept: 0.999877
Slope: -2.41086
Accuracy: 0.784553
Sensitivity: 0.763514
Specificity: 0.816327

-------Confusion Matrix-------
                    Predicted Survived  Predicted Dead
Actually Survived   113                 35
Actually Died       18                  80

(Training took 55 microseconds)
Program terminated.
```



```
Opening file titanic.csv
Reading line 1
heading: "","pclass","survived","sex","age"
new length: 2000
Closing file titanic.csv
Number of records: 1046

-------Naive Bayes-------

-------Apriori Probabilities-------
countSurvived: 312
p_survived: 0.39
p_died: 0.61

-------Conditional Probabilities-------
pclass:
            1           2           3
Survived    0.416667    0.262821    0.320513
Died        0.172131    0.22541     0.602459
sex:
            1           2
Survived    0.679487    0.320513
Died        0.159836    0.840164
age:
            1           2
Survived    28.8261     14.4622
Died        30.4182     14.3231

-------Raw Probabilities-------
    Survived    Died
[0] 0.444967    0.555033
[1] 0.0774705   0.922529
[2] 0.471226    0.528774
[3] 0.91386 0.08614
[4] 0.564037    0.435963
[5] 0.0738416   0.926158
[6] 0.0671139   0.932886
[7] 0.562947    0.437053
[8] 0.162245    0.837755
[9] 0.124127    0.875873

Accuracy: 0.735772
Sensitivity: 0.626087
Specificity: 0.832061

-------Confusion Matrix-------
                    Predicted Survived  Predicted Dead
Actually Survived   72                  43
Actually Died       22                  109

(Training took 317 microseconds)
Program terminated.
```

The first noticeable difference is the time it took to train each model. Logistic Regression was considerably faster than Naïve Bayes (55ms vs 317ms). Considering that this is a small training set of 800 observations, with more training data, this difference in speed could be significantly greater. In terms of accuracy they came out to be similar but Logistic actually gave a greater accuracy in this case. This is probably because while they have similar specificity values, the Naïve Bayes model had a much lower sensitivity, meaning there was more false negatives. Seeing that Bayes also had a higher (slightly) specificity, this means the model was just predicting more deaths in general. Typically, Bayes should be more accurate than this (for this kind of data), so it's possible the training data was biased in some way, or the manually created function was faulty. (I also compared the results from Logistic Regression to the built-in functions in R and its off by one prediction, most likely due to rounding error)

These two model types are a good example of generative vs discriminative classifiers. Logistic Regression is uses discriminative classifiers and Naïve Bayes uses generative ones and obviously are both used for classification.

Where they differ is *how* they estimate with the data. Discriminative Algorithms directly estimate the parameters for P(Y|X). Whereas Generative Algorithms estimate the parameters for P(Y) and P(X|Y). This makes generative algorithms more biased, and less varying, but worse when it comes to larger sets of data.

Reproducible research is not unique to Machine Learning. In every field of science/technology, it's important to keep good documentation and available resources so that others may try to recreate what was done. This is so other people can confirm the validity of the results and can continue to build off of it. With Machine Learning (or any code-based research), one of the easiest ways to practice this is by publishing the code. Especially using some form of source control (like Git), you can show the results through every stage of the research (if the model or the dataset evolves). This pushes the field forward as researchers continue to share and develop ideas together.

# Works Cited

"Reproducible Research for Scientific Computing: Tools and Strategies for Changing the Culture," *Reproducible Research for Scientific Computing*, 2012.

Z. Ding, "5 - reproducibility," *Machine Learning Blog | ML@CMU | Carnegie Mellon University*, 24-Aug-2020. [Online]. Available: https://blog.ml.cmu.edu/2020/08/31/5-reproducibility/. [Accessed: 04-Mar-2023].