Sharmin Gaziani
CS 4375.004

ML Algorithms from Scratch

a. Runs of my code:

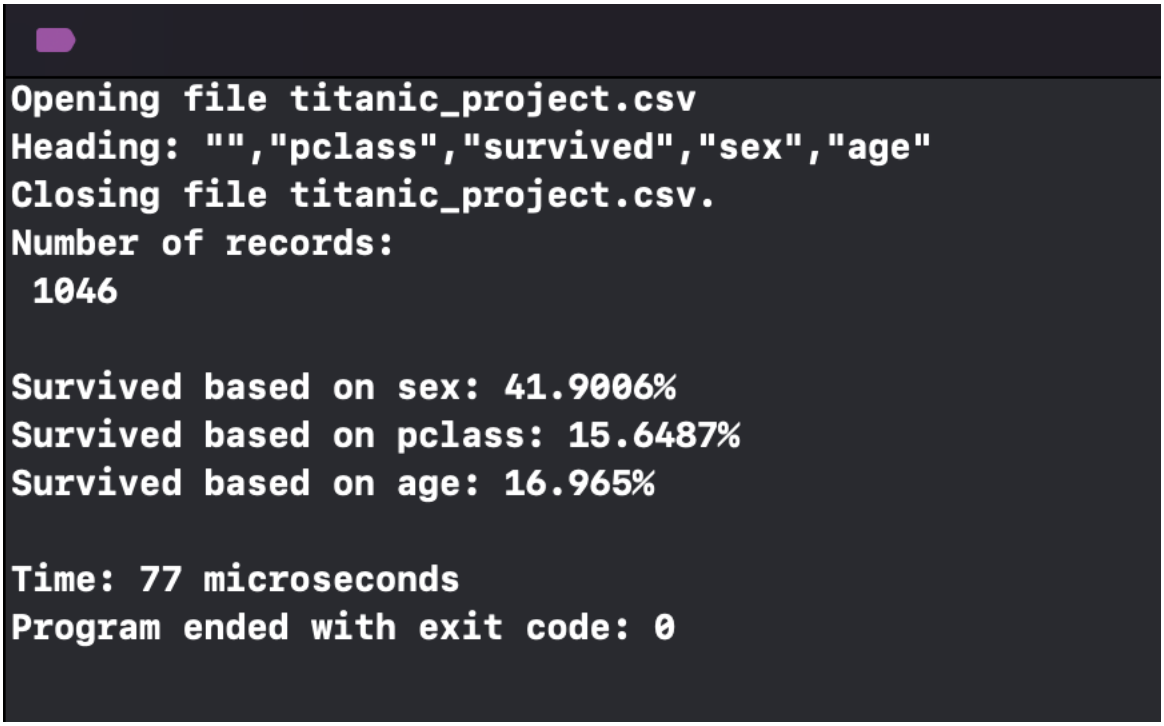
Program 1:

```
Opening file titanic_project.csv
Heading: "","pclass","survived","sex","age"
Closing file titanic_project.csv.
Number of records:
 1046

coefficients:
w0 = 1
w1 = -2.2

Test metrics:
Accuracy: 66.6667%
Sensitivity: 0.566
Specificity: 0.566
Time: 75 microseconds
Program ended with exit code: 0
```

Program 2:



b. Analyze the results:

After doing running the same algorithms in R, I will say that doing this from scratch in C++ is a bit more challenging. It also took longer when programming the algorithms. For my first program, I predicted survived based on sex. I first did my train and test sets and started running the logistic regression in C++. I noticed that my accuracy was not too high for the first program and that the sensitivity and specificity was pretty much the same for both variables. My algorithm did not take long either, it took 75 microseconds which is a short amount of time. I also outputted my w0 and w1 coefficients to the console as well. The logistic regression was easy to implement and was very efficient with the amount of data I was using. For my second program, I did something similar, but I found the survived based on age, pclass, and sex using Naïve Bayes. I outputted the percentage of survived based on sex, pclass, and age. I noticed with my calculations, the percentage was not very large where pclass and age was 15% and 16%. I also showed the time for my algorithm which took 77 microseconds, similar to program 1. I think using Naïve Bayes has many benefits because it was easy to implement, and it was able to use both continuous and discrete data.

c. Generative classifiers vs. discriminative classifiers

A discriminative classifier refers to Logistic regression that is directly estimated by classifiers from P(Y|X). On the contrary, generative classifiers are estimated by Naïve Bayes using the parameters P(Y) and P(X|Y). Both of these classifiers have similar functionalities as well as some differences between them. Both of their parameters are

directly from training data and the parameters assume the function form of either P(X|Y) or P(Y|X). Although both learn from the same training data, often times, the models are going to learn different probabilities from the set.

Some ways generative differs from discriminative is that discriminative will model the decision boundary between the classes. The approach for this model includes assuming the functional form and estimating the parameters from the training data. Generative, however, will model the actual distribution of each class. When approaching this mode, we find the conditional probability by estimating the prior probability and the likelihood probability. This is done with the training data and Bayes Theorem. Some examples of generative classifiers are Hidden Markov Models, Naïve Bayes (as stated before), and Bayesian networks. Examples for discriminative includes Logistic regression, as earlier stated, and nearest neighbor.

Sources:
https://medium.com/@mlengineer/generative-and-discriminative-models-af5637a66a3
https://www.analyticsvidhya.com/blog/2021/07/deep-understanding-of-discriminative-and-generative-models-in-machine-learning/#:~:text=Difference%20Between%20Discriminative%20and%20Generative%20Models,-Let's%20see%20some&text=In%20mathematical%20terms%2C%20discriminative%20machine,P(X%2C%20Y).

d. Reproducible research in machine learning is important because we learn from others through data sharing. If results are not reproducible by others, we cannot learn from the experiment and innovate within that idea. Within machine learning, we are constantly learning; from computers, from ourselves, from others, and from research. From learning, we define new challenges, interpret new barriers, reduce current barriers, and provide experiments with new research. According to the article “Reproducible Research for Scientific Computing: Tools and Strategies for Changing the Culture”, it is important for research to be replicable, run a code and get the same results, and reproducible, create code to independently verify published results. I agree with both statements, but I believe results being reproducible is more important than the latter. There are multiple ways to approach the same problem while getting the same results as well. Although some may be more efficient than others, the results should not vary.

When doing experiments, it is important to do the test multiple times to see how results vary. There can be a number of reasons why someone gets the results they get and one of those reasons are often times human error. If multiple people run the same test and get different results, it is not an accurate test and there is not much we can learn from the results. But, if multiple people run the same experiment and get the same results, we can learn from this experiment and make changes to it. We can find different approaches to the problem in attempt to get the same result, we can make the algorithm/ experiment more efficient, and find ways to improve it.

Machine learning has grown tremendously due to the fact that results have been reproducible based on the proposed standards as data. The studies do not help future scientists when results cannot be reproduced, or improper documentation is published. Reproducibility can be implemented by publishing how the experiment went, the code used, the results, and how many trials it took. When not everything from the study is released, no matter how accurate the results are, the results are not reproducible.

Sources:
https://staff.washington.edu/rjl/pubs/cise12/CiSE12.pdf
https://www.nature.com/articles/s41592-021-01256-7#citeas