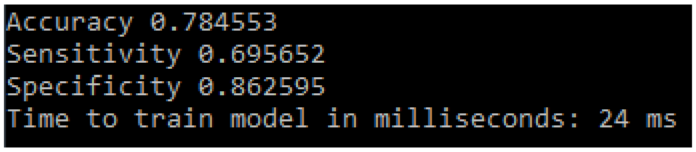
Logistic Regression and Naïve Bayes

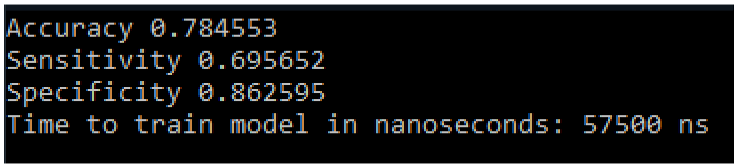
Subash Chandra and Derrick Martin

Logistic Regression.cpp



```
Accuracy 0.784553
Sensitivity 0.695652
Specificity 0.862595
Time to train model in milliseconds: 24 ms
```

Naïve Bayes.cpp



```
Accuracy 0.784553
Sensitivity 0.695652
Specificity 0.862595
Time to train model in nanoseconds: 57500 ns
```

Both Algorithms ended up predicting exactly the same thing for the dataset, but the Naïve Bayes algorithm ended up being nearly 400 times faster, even though it is using all the variables as predictors as opposed to Logistic Regression which is only using sex as the predictor. This seemed suspicious, but it was verified in R, and we arrived at the same conclusions.

Generative vs Discriminatory classifiers

Generative Classifiers work by analyzing the data to form prior "likelihood probabilities" for events, and then observes the posterior probability that an event occurred after taking all variables into account. Discriminative Classifiers on the other hand work by simply trying to split the dataspace into clusters with lines, and then calculating the probability by finding the distance from any point to the lines to find out how likely it belongs there.

There are Pros and Cons to both of these. Generative Classifiers, for example are able to deal with missing data much better, because it is able to model the data and find the missing value. Discriminative algorithms are much worse, and will naively classify the data based on its position on a graph. This leads to the issue where Generative algorithms are much worse at dealing with Outliers because it will try to fit its models to the data and make strong assumptions, no matter how convoluted the data is. Discriminative algorithms are much better at dealing with outliers because it just effectively minimized/ignores the data points that are too far from the lines it draws.

Reproducible Research

An experiment is “reproducible” if the results of the experiment are able to be replicated by any researcher as long as they have access to the same data, tools and computing power. Reproducibility is as important to Machine Learning as it is to most other sciences. One of the most important reasons is trust. I don’t normally trust Machine Learning models because I know they are imprecise and inaccurate in real world use. If I was making a machine learning application that gave different results depending on arbitrary events, I would not trust that model to be able to actually solve real world problems. Another important reason is that it brings the entire scientific community forward. Everything we know has been built on centuries of reproducible experiments passing down from our ancestors, and to be able to put my own footprint in that sand is quite exhilarating.

Reproducibility is not a monolithic feature. It requires care and precision. Here are some ways to implement reproducibility into your own ML project :-

1) Randomness – Using randomness to make train/test splits is quick and easy, but it does not allow for reproducibility. Instead using the first X for training, and the rest for testing will make the data reproducible. Alternatively, you can set a seed to allow the same random numbers to show up every time.
2) Floating point precision – Sometimes unnecessary truncation or rounding can be the deciding factor.
3) Version Control – If changes in the code/data are not properly tracked and logged, this too can cause a reproducibility issue.
4) Accuracy through Volume – If these cannot be dealt with, then ultimately the choice might just be to run the model a bunch of times and try to draw statistical conclusions from the aggregate of results rather than just one.

Sources –

A Guide to Computational Reproducibility in Signal Processing and Machine Learning - https://arxiv.org/pdf/2108.12383.pdf

Reproducible Machine Learning - https://towardsdatascience.com/reproducible-machine-learning-cf1841606805

Decisive Edge - https://www.decisivedge.com/blog/the-importance-of-reproducibility-in-machine-learning-applications/