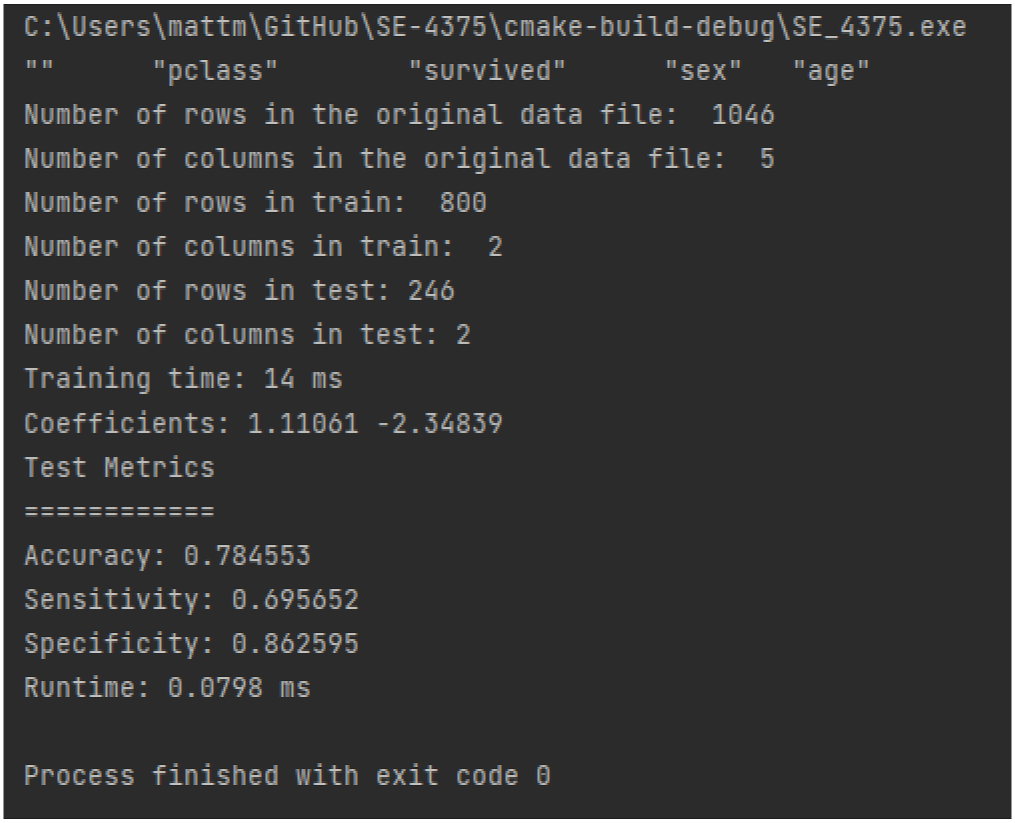
Dmitrii Obideiko
Matt McCoy

ML Algorithms from Scratch - Report

Logistic Regression Analysis



```
C:\Users\mattm\GitHub\SE-4375\cmake-build-debug\SE_4375.exe
""      "pclass"        "survived"      "sex"   "age"
Number of rows in the original data file:  1046
Number of columns in the original data file:  5
Number of rows in train:  800
Number of columns in train:  2
Number of rows in test: 246
Number of columns in test: 2
Training time: 14 ms
Coefficients: 1.11061 -2.34839
Test Metrics
============
Accuracy: 0.784553
Sensitivity: 0.695652
Specificity: 0.862595
Runtime: 0.0798 ms

Process finished with exit code 0
```

Coefficients: 1.11061 -2.34839

There are two coefficients: 1.11061 and -2.34839. The first coefficient represents the bias term, and the second coefficient represents the weight or importance of the single input feature in the model. These coefficients were learned by the algorithm during training and were used to make predictions on the test data.

Accuracy: 0.784553

The accuracy is 0.784553 or 78.46%, which means that the model correctly predicted the outcome of 78.46% of the observations in the test data.

Sensitivity: 0.695652

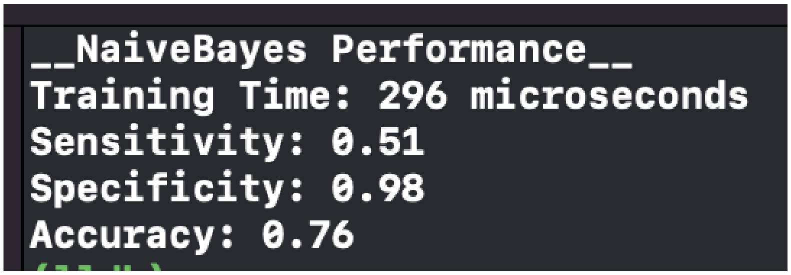
The sensitivity is 0.695652 or 69.57%, which means that the model correctly predicted 69.57% of the actual positive cases in the test data.

Specificity: 0.862595

The specificity is 0.862595 or 86.26%, which means that the model correctly predicted 86.26% of the actual negative cases in the test data.

Runtime: 0.0798 ms

NaiveBays Analysis



```
__NaiveBayes Performance__
Training Time: 296 microseconds
Sensitivity: 0.51
Specificity: 0.98
Accuracy: 0.76
```

The NaivBays Analysys perforated fairly well, with an accuracy of 0.76. It seems that the algorithm that I developed is good for predicting if the person died as a result of the titanic's crash the specificity is fairly high which is .98; however, it is not as accurate when it comes to predicting if the person survived as the sensitivity is just .51

Generative Classifiers vs Discriminative classifiers [1]

Generic classifiers and discriminative classifiers are machine-learning algorithms that are both used for classification tasks. Generative classifiers use the joint probability distribution and their primary focus is on finding how features and target variables become related to each other. After some exposure to data, general classifiers are capable of making predictions with the new data that they receive. Discriminative classifiers, on the other hand, always try to find boundaries that separate classes.

Unlike discriminative classifiers, generative classifiers are less resistant to outliers as they could greatly affect the distribution. If a discriminative classifier encounters an outlier, it would just make it a misclassified example. Generative classifiers also need more data to create accurate distributions, unlike discriminative classifiers which don't need as much data. In addition, generative models require more computation, thus they are more computationally expensive than discriminative models.

Reproducible Research in Machine Learning

Reproducible Research in Machine Learning is the ability to run a certain algorithm on various datasets and always get either the same or similar results on the project that you are working on. When it comes to machine learning, the results that get produced aren't always the same, which means that machine learning projects aren't always reproducible.

Reproducibility plays a huge role when it comes to continuous changes in the software as well as the delivery cycle. It increases data consistency which can definitely help when projects move from development to production. In addition, reproductivity increases trust and credibility with any machine learning project. It displays that the machine learning project was well-designed and built [3]. It also helps just make any improvements with machine learning projects as we can compare them with the previous versions and prove that there was some improvement made. It can also give us more proof of correctness as if you get different results all the time, it is usually a bad sign. In addition, it can reduce any randomness in general as scientists won't think that it was caused due to a particular parameter [4].

One of the most important things when it comes to implementing reproducibility into machine learning is that the project keeps proper documentation. Good documentation reduces ambiguity, which reduces the likelihood of successfully executing the project repeatedly. Another thing that developers could do to increase the likelihood of repeated successful execution is to use "the pipeline mentality". For example, developers could create sequential models that can perform some common tasks like reading data or clearing data that could be regardless if there are some changes that are needed to be made in the project [3].

Sources cited

[1] Yıldırım, Soner. "Generative vs Discriminative Classifiers in Machine Learning." Medium, Towards Data Science, 14 Nov. 2020, https://towardsdatascience.com/generative-vs-discriminative-classifiers-in-machine-learning-9ee265be859e.

[2] "The Importance of Reproducibility in Machine Learning Applications." *DecisivEdge*, 7 Dec. 2022,

[3] 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.

[4] Ding, Zihao. "5 - Reproducibility." *Machine Learning Blog | ML@CMU | Carnegie Mellon University*, 24 Aug. 2020, https://blog.ml.cmu.edu/2020/08/31/5-reproducibility/.