a) Logistic Regression:

Opening file
Reading line 1
Headers: "","pclass","survived","sex","age"
weights: 0.999877
weights: -2.41086
accuracy: 0.784553
sensitivity: 0.695652
specificity: 0.862595
learning rate: 0.001
iterations: 5000
training time(microseconds): 6963776

Naïve Bayes:
Opening file
Reading line 1
Headers: "","pclass","survived","sex","age"
Prior probability:
Survived: 0.39
Dead: 0.61
train_dead: 488
train_survived: 312
p1s0: 0.172131
p1s1: 0.416667
p2s0: 0.22541
p2s1: 0.262821
p3s0: 0.602459
p3s1: 0.320513
lh_pclass:
0.172131 0.22541 0.602459
0.416667 0.262821 0.320513
lh_sex:
0.159836 0.840164
0.679487 0.320513
age_mean:
30.4182 28.8261
age_var:
14.3231 14.4622
Raw: 1.9496e-39 1
Raw: 0.963097 0.0369027
Raw: 1.75387e-39 1
Raw: 0.171213 0.828787
Raw: 0.616074 0.383926

accuracy: 0.47561
sensitivity: 0.330435
specificity: 0.603053
prediction: 246
training time(microseconds): 974

b) The data shows that logistic regression was better at predicting whether or not someone survived compared to Naïve Bayes. This could have also been affected by the fact that Naïve Bayes had other predictors that could have cluttered its predictions with data that was not as correlated as gender was. In fact, gender seems to have a strong correlation between living or dying, as women were much more likely to live than men were, and thus the logistic regression had an easier time predicting the target. Age and class seemed to have much less of a correlation with living or dying and thus the naïve bayes could not predict the outcome as easily as logistic regression.

c) Generative classifiers and discriminative classifiers are both very important and useful when it comes to machine learning. Generative classifiers are useful for figuring out how features and target variables occur together. Finding out how the data is generated allows this classifier to make more accurate predictions on the target based on how data was generated in the past and using that as a prediction for how it will be generated now. On the other hand, discriminative classifiers look at what makes the different features different from one another to predict the target. Looking at these differences and adjusting whether the boundaries between the data is hard or soft can modify how the data is viewed and predicted.

Both of the two different types of classifiers are useful in their own ways and fills their own niche in the machine learning world. Generative models take more space in the computer to run due to needing to see the entire distribution and all of its data in order to make accurate predictions. Discrimination can handle outliers better than Generative can and are generally less taxing on the machine that is running it. Both models are very useful in their own ways but have a separate use depending on the data that needs to be analyzed. Some situations call for one and another may call for the other, but by being familiar with both we can use the correct one for the job at the time.
Source: https://towardsdatascience.com/generative-vs-discriminative-classifiers-in-machine-learning-9ee265be859e

d) Reproducibility in machine learning is something that is highly sought after. Being able to run your algorithms multiple times and continually get the same results means that it is much easier to analyze and debug the program, as well as ensure that the program is something that is consistent and will not only return good results occasionally. [1] Clients want to see a program that is reliable and having one will make your business much more successful. It creates a sense of trust between the program, the programmers, and the clients when the finished product is something that is consistent and reproducible.

There are many different ways to create reproducible data and as long as the same data/program/tools are used, it will still be considered reproducible. [2] Studies have been done on the different reasons why machine learning would not be considered

reproducible, whether it be that sometimes it takes more space or takes longer to process, but there have been breakthroughs on different ways to keep it consistent. A big factor is to have a deep understanding of the programs space and time complexity to understand their best and worst case scenarios and when those scenarios may occur. [2] Making the code and data easily accessible is another hugely important factor in making code reproducible. Ensuring that the source code and the raw data is readily available ensures that others that try your experiments and tests can reproduce it with the same results, or at least nearly the same, and allows for more widespread use of the program.

Another problem that needs to be addressed when talking about reproducibility in machine learning is the fact that data can be easily manipulated or misrepresented and show incorrect findings, and whether intentional or not, it is a large issue in the field. [2] Overcoming this problem requires very clear and open communication about how the data was acquired, and how it is being displayed and distributed. Allowing other people to download the code and the data ensures that others can fact check and explore the data to their liking, ensuring that the data cannot be misrepresented.

[1] 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.

[2] https://blog.ml.cmu.edu/2020/08/31/5-reproducibility/