The provided code analyzes a dataset containing 569 breast cancer diagnoses. The data is formatted into data frames using the Pandas Python package. The information is split into a new data frame that breaks out the diagnoses by the outcome. 212 of the cases were found to be malignant, while 357 were benign.

A random forest algorithm is used to train a model. The standard 80-20 split was used for training and test data. Five folds are used in the k-fold cross-validation technique. The final output shows an accuracy of 95.16% and a standard deviation of 0.033. The precision was found to be 95% for benign cases and 96% for malignant cases. This is caused by 2 false positives and 3 false negatives.

Overall, the accuracy report of this model was good. The issue comes in when considering the use of the model. A false positive on a cancer screening can negatively impact someone’s life, but they should ultimately be alright due to not having cancer. A false negative is a bigger issue.

Someone would most likely go on with their life as if they did not have cancer. This would leave the cancer to grow unhindered. This diagnosis would lead to the person having a much shorter life span due to a misdiagnosis.

There are many ways to tweak the parameters of the model. The number of folds could be increased. The n-estimators could be changed from the given “10, 50, 100, 200” values. The random state could be massaged in the random forest classifier instantiation function call. Other algorithms could be tried. Such as nearest-neighbor, SVM, and logistic regression.

Machine learning has helped the human race get further faster. However, there are still areas in which we should not blindly trust the output of algorithms. Cancer detection is one such area. A second opinion given by a human will help reduce the number of false positives and, more importantly, false negatives.