Ben van Zyll
CPSC393 — 01
DUE: 5/21/22

Breast Cancer Diagnosis
Binary Classification

Final Report

*All code can be accessed through Jupyter Notebook files in the Final Project folder

ABSTRACT

Breast cancer is one of the most plaguing forms of disease in the world, with 1 in 8 American women developing some form of invasive breast cancer at some point in their life[1]. Furthermore, among all forms of newly diagnosed cancer in women, breast cancer is responsible for a whopping 30 percent[2]. In developed countries, only 70 percent of breast cancer diagnoses are in the first or second stage, while the other 30 percent are at even higher mortality risk, being diagnosed in the third or fourth stage[2]. In developing countries, the stage one and two diagnosis rate drops to a mere 20 percent, making the other 80 percent cases practical death wishes — at the third or fourth stages with already poor healthcare infrastructure[2].

The most common, state-of-the-art breast cancer diagnosis method is the mammogram, while other more uncommon methods include MRI-guided biopsies, ultrasound-guided biopsies, among some others. Upon further investigation, we came across an article from the academic journal *Informatics in Medicine Unlocked*, where machine learning engineers attempted to use a logistic regression model to predict breast tumor diagnoses. The model yielded very good results — an accuracy score of about 95.71 percent, a sensitivity of about 99.44 percent, and a specificity of about 83.33 percent. Our goal was to experiment with different machine learning algorithms to create three baseline models — logistic regression, random forest, and support vector machine — as well as a deep learning implantation — a feed-forward neural network — to improve upon the model created by *Informatics in Medicine Unlocked* and introduce a scalable, cheap, and accurate means of cancer diagnosis.

EXPLORATORY DATA ANALYSIS

The Breast Cancer Dataset is an open-source, publicly available dataset that we found on Kaggle. This data contains a total of 569 observations of breast tumors and contains 2 different variables, one of which is extraneous in terms of model building — it just stores a unique ID value — and one of which is going to be the label that we ultimately predict, which is the diagnosis of the tumor (malignant or benign). Feature variables in this dataset include but are not limited to:

- radius_mean
- perimeter_mean
- area_mean
- texture_mean
- radius_worst
- perimeter_worst
- area_worst
- symmetry_worst
- fractal_dimension_worst

Figure 1

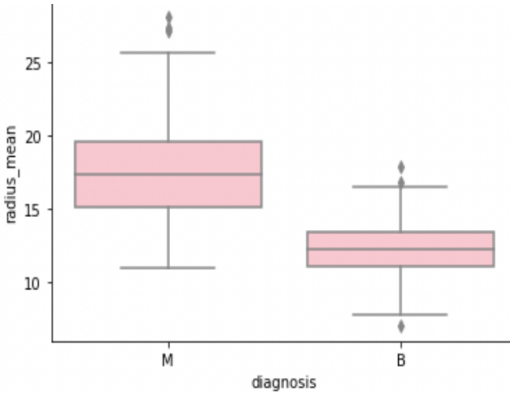
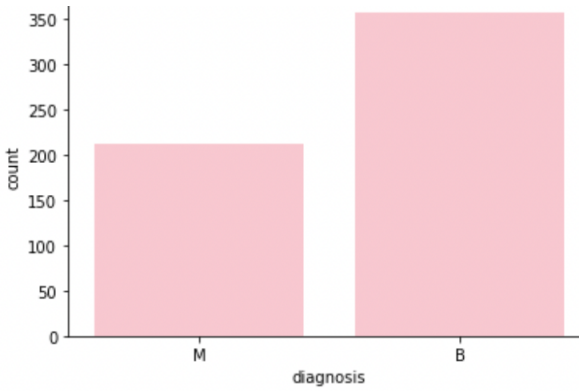


Figure 2



The figures above give us into the relationship between mean radius and diagnosis (left) — as we can see with a malignant diagnosis skewing toward larger mean radius values. We could make a similar plot with all feature variables to observe the relationships of that feature variable on the diagnosis outcome. The figure on the left show us the overall distribution of diagnosis outcomes in the data, with about 150 more benign diagnoses than malignant.

Figure 3

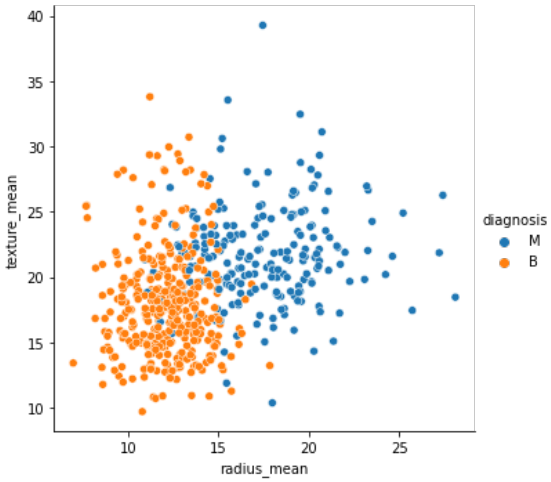


Figure 4

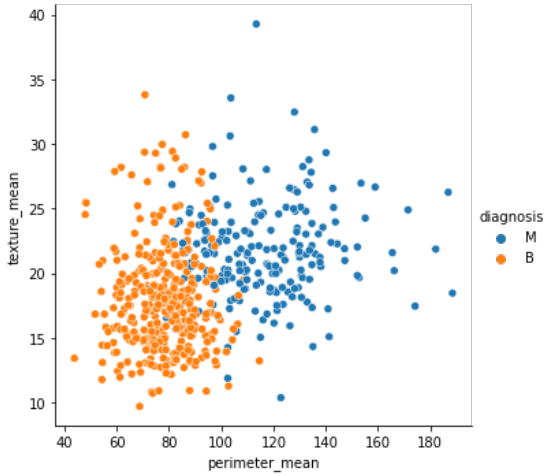


Figure 5

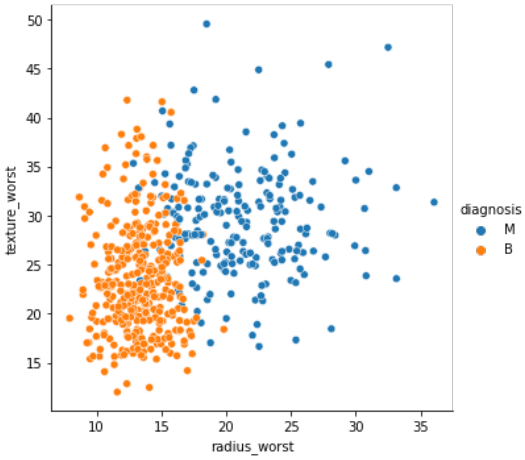
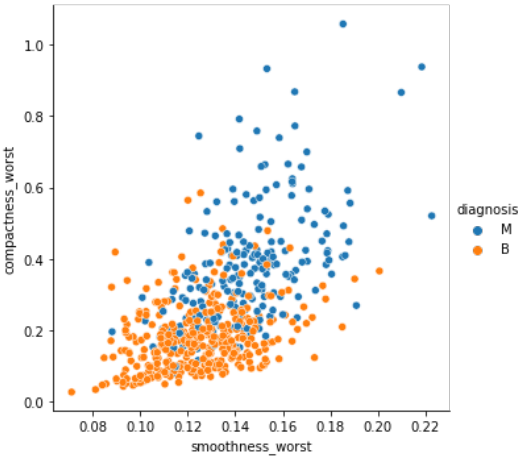


Figure 6

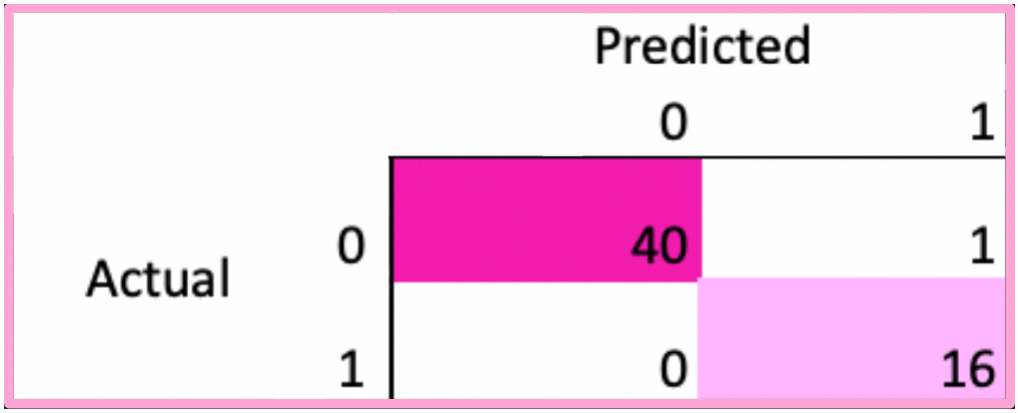


The figures above give us quite a bit of insight into how a model may perform. The four scatterplots above plot continuous feature variables against one another, and color code each observation based on its diagnosis — blue if malignant and orange if benign. To begin, we see that typically, tumors with more extreme values, like a greater radius, perimeter, texture, and compactness are more likely to be malignant. What is more telling however, at least from a model performance standpoint, is that the outcome data appears to be very linearly separable. Our hypothesis after seeing this pattern was that very simple baseline models may perform very well, and we will get into the results of those baseline models now, starting with the logistic regression model.

BASELINE MODEL I — Logistic Regression

Our baseline modeling methodology remained fairly consistent across the three models built, so we will only deep dive into the technical processes this once. After importing necessary libraries, step one of model building process was to drop the irrelevant "id" column, which added no meaning or value to any observations. We then one hot encoded the binary outcome variable — 1 for malignant, 0 for benign — so that the data could be interpreted by the algorithm. We then split the data into training, validation, and testing sets. Next, we utilized GridSearchCV as well as a function that we created to display how logistic regression performed given a set of different values for the C hyperparameter. The C hyperparameter is a regularization mechanism, as it applies a penalty for error during training — a higher C value meaning more penalty and more likely to overfit to the training data. The logistic regression model yielded optimal C values of 1,000, 100, and 10 out of seven different values, so we then tested those values of C on the validation data. All three values of C yielded the same results when tested on the validation set, so we went with the value of 10 as it provided us with just as good of results, and more generalizability. After testing the logistic regression model built with a C value of 10 on the testing set, we ultimately saw the following results:

Accuracy: 0.982 Precision: 0.941 Recall: 1.0



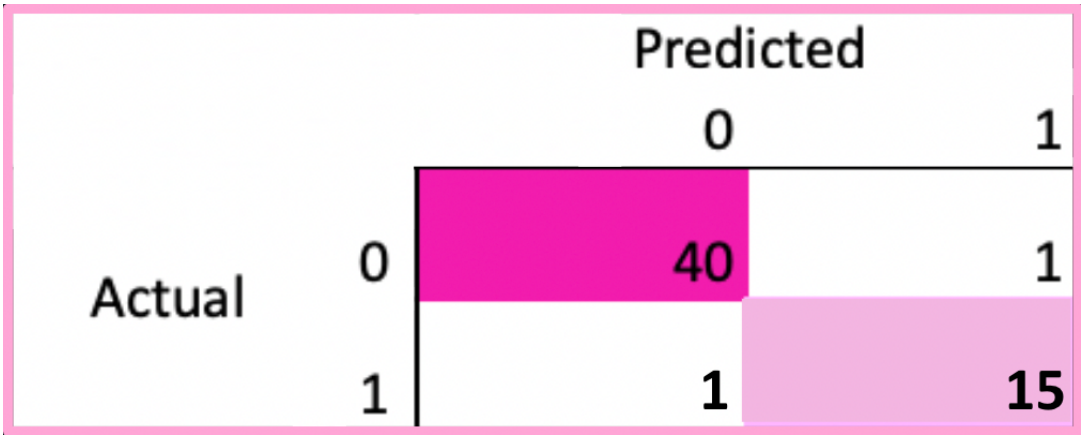
The confusion matrix above gives us insights into where the predictions were classified exactly, and the results are brilliant; we see only one misclassification, and it was a false positive, much less costly than a false negative in this context. Ultimately, the logistic regression model performed nearly perfectly.

BASELINE MODEL II — Random Forest

The next baseline model that we decided to build was a random forest classifier. Again, methodology was all the same, except when we got to GridSearchCV and hyperparameter tuning. Instead of tuning a C value, we had to tune n_estimators and max_depth. Skipping the technicalities, we ultimately concluded that the optimal hyperparameters of the model were an n_estimators value of 100 and a max_depth value of 8. After testing these hyperparameters on the testing set, we ultimately saw the following results:

Figure 8

Accuracy: 0.965 Precision: 0.938 Recall: 0.938



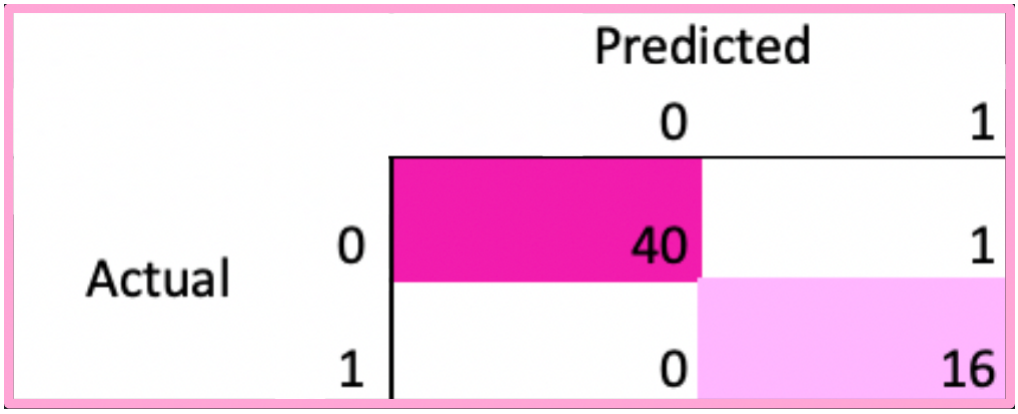
The confusion matrix above gives us insights into where the predictions were classified exactly, and the results are nearly as good as the logistic regression's, with a small caveat; we see only one extra misclassification, but it was a false negative, which would be much more costly in this context. Someone with breast cancer would be told that they do not have it. One can imagine the repercussions. Therefore, this model, though performing very well, is not as impressive as the logistic regression model.

BASELINE MODEL III — SVM

The last baseline model that we decided to implement was an SVM classifier. For the third time, all the methodology was all the same, except when we got to GridSearchCV and hyperparameter tuning. In this case, we tuned the C hyperparameter again, as well as the kernel hyperparameter, which simply refers to the type of kernel trick used, such as linear, sigmoid, Gaussian RBF, and more. Skipping the technicalities again, we ultimately concluded that the optimal hyperparameters of the model were a C value of 1 and a linear kernel trick implementation. After testing these hyperparameters on the testing set, we ultimately saw the following results:

Figure 9

Accuracy: 0.982 Precision: 0.941 Recall: 1.0



The results of our SVM classifier were identical to the results of the logistic regression model. To summarize the analysis from that confusion matrix, the results are brilliant; we see only one misclassification, and it was a false positive, much less costly than a false negative in this context. Ultimately, the logistic regression model performed nearly perfectly. In the end, the baseline models performed brilliantly, with the logistic regression and SVM classifier equally producing the best results, with the random forest classifier not far behind.

DEEP LEARNING — Feed-Forward Neural Network

For our deep learning implementation, we went with a standard feed-forward neural network. Like all other models, we began removing the notorious “id” variable. The overall architecture of our model is as follows:

- Overall structure: Sequential model with two dense layers
- First dense layer: 32 units, hyperbolic tangent (tanh) activation
- Second dense layer: 1 unit, sigmoidal activation
- Optimizer: rmsprop
- Loss function: binary_crossentropy
- Metrics: accuracy
- L1 Regularization

We decided to go with a much simpler model, as our dataset is very simple in itself, with visibly linearly separable data and only 569 total observations. We made sure not to overcomplicate the model for this reason, so we only included two dense layers. As for our optimizer, we implemented rmsprop, or root mean squared error propagation, which is an extension of gradient descent that utilizes a decaying average of partial gradients in the adaptation of the step size for each parameter. We also used L1 regularization for model generalizability. We implemented binary cross entropy as our loss function as this is a binary classification problem, diagnosing tumors as either malignant or benign. Additionally, because this is a classification problem, we implemented accuracy for our evaluation metric, as an accuracy score is a holistic evaluation of a model’s ability to correctly classify.

As for our neural network’s overall performance, it was poor compared to the success of our baseline models. See below for graphical representations and explanations of accuracy and loss of the model over 20 epochs.

Figure 10

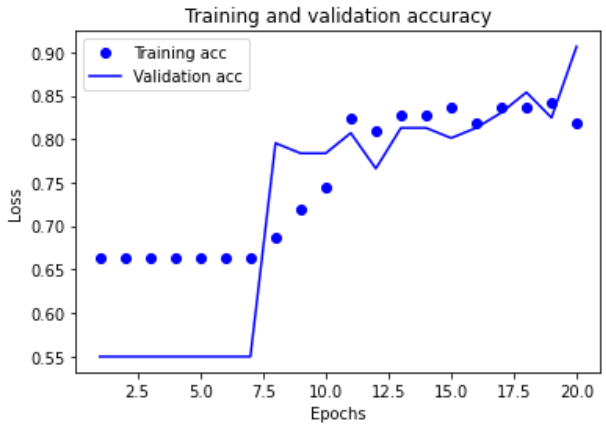
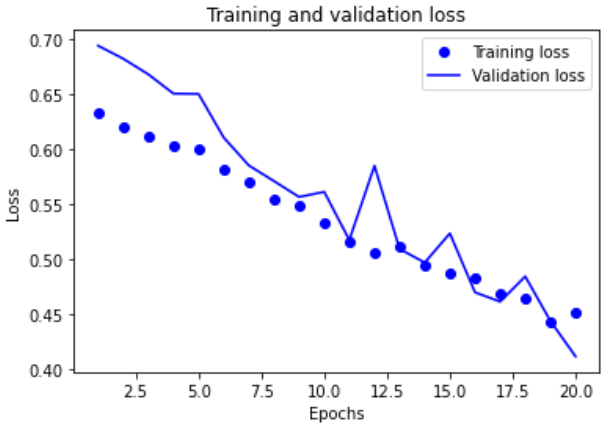


Figure 11



As seen in the figures above, we that towards the last few epochs, training and testing accuracy seem to converge, and then an ultimate spike in testing accuracy at the twentieth epoch — which we will write off as an anomaly. Nonetheless, our model did not overfit to the training data, likely thanks to the L1 regularization we applied as well as the simplicity of the model on a simple dataset. While we would normally determine a model to be quite impressive with a testing accuracy of about 0.906, the very basic baseline models proved more capable of classifying, despite a neural network having a far more powerful architecture. Ultimately, our feed-forward neural network did disappoint, as it was severely outperformed by all three baseline models, but we sort of expected such a result after the exploratory data analysis discussed earlier.

FINAL THOUGHTS & FURTHER EXPERIMENTS

To conclude, the logistic regression and SVM models yielded the most impressive results, with the random forest classifier one misclassification behind and our deep learning implementation drastically underperforming for such a powerful model. Some thoughts we have as to why the baseline models may have performed better are that perhaps the neural network was simply poorly configured and designed, and we the human practitioners are to blame, and that the simplicity of the baseline models are simply better fit for the simplicity of the data.

In terms of future experiments, for the logistic regression and SVM classifier to truly be effective means of classifying tumors, we would need some sort of widely deployable, cheap, accessible technology that would gather the statistics of a tumor that we use (i.e. mean radius of the tumor, worst radius of the tumor, etc.) so that that information can be fed into the model and it can determine a diagnosis. Without a technology like that, the model is somewhat obsolete. It could also be interesting to include a range of other data in future experiments, not only pertaining to the tumors themselves, but data regarding people's lifestyles as well, such as age, income level, insurance, diet, and much more. Lastly, with further model development, we would need to ensure that we maintain the bias against false negatives. Classifying someone's tumor as benign when it is malignant is extremely more harmful than classifying someone's tumor as malignant when it is benign. Our logistic regression and SVM classifier models did a great job of this, but it is definitely a major factor to consider in future experiments.

REFERENCES

Challenges to the early diagnosis and treatment of breast cancer in developing countries. Karla Unger-Saldaña. World J Clin Oncol. 2014 Aug 10; 5(3): 465–477. Published online 2014 Aug 10. doi: 10.5306/wjco.v5.i3.465

Breast Cancer Facts and Statistics. BreastCancer.org. 2022 March 10; Published online 2022 March 10. https://www.breastcancer.org/facts-statistics

Laila Khairunnahar, Mohammad Abdul Hasib, Razib Hasan Bin Rezanur, Mohammad Rakibul Islam, Md Kamal Hosain, Classification of malignant and benign tissue with logistic regression, Informatics in Medicine Unlocked, Volume 16, 2019, 100189, ISSN 2352-9148, https://doi.org/10.1016/j.imu.2019.100189.
(https://www.sciencedirect.com/science/article/pii/S2352914818301497)