# Question 1

A graph of blue bars

Description automatically generated with medium confidenceA graph with a blue line

Description automatically generatedThis question involved creating a binary classifier to predict whether someone was positive for diabetes. The methodology for this was the same as the gradient descent algorithms from the past problems. After calculating the predictions (values between 1 and 0), we simply call predictions below 0.5 negative and above 0.5 positive for diabetes. If this diabetes dataset is normalized, it makes the model freak out and produces unreliable results. But using standardization effectively solves this issue and produces a good model.

Regularization was also applied to try and remove the parameters that don’t have as much of an effect on diabetes, such as blood pressure and skin thickness. It emphasizes glucose levels and BMI, which are historically the most important factors in predicting diabetes.

|  |  |  |
| --- | --- | --- |
| Confusion Matrix | Positive (1) | Negative (0) |
| Positive (1) | (true positive) **122** | (false positive) **33** |
| Negative (0) | (false negative) **11** | (true negative) **22** |

Accuracy: 78.7097% | Precision: 91.73% | Recall: 84.72% | F1: 0.8809

These are pretty good results for a classifier model. The model gets about 80% of predictions correct, and about 92% of the cases that we predicted as positive were positive, which is an encouraging result. The F1 score combines the previous scores (recall, precision) to give us a view into how well the model is performing. Generally, this model is really good, but it could use improvement with its recall score.

# Problem 2

This problem is a repeat of the previous problem. Without L1 regularization, the model performs quite well.

A graph with a blue line

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Confusion Matrix | Positive (1) | Negative (0) |
| Positive (1) | (true positive) **112** | (false positive) **4** |
| Negative (0) | (false negative) **2** | (true negative) **2** |

Accuracy: 96.5517% | Precision: 98.25% | Recall: 98.25% | F1: 0.9825

This is an extraordinarily well-performing model, and oddly enough most of the data predicts that 0th option (malignant). Adding L1 regularization bumps up all the scores a little bit and improves the training speed.

A graph with a blue line

Description automatically generatedConfusion Matrix:

[113, 3]

[2, 1]

Accuracy: 97.4138% | Precision: 98.26%

Recall: 99.12%

F1: 0.9869

# Problem 3

Instead of using the gradient descent algorithm from last time, we now switch to a naïve Bayes classifier. This algorithm works on the principle of Bayes’ theorem and is considered naïve because it makes an assumption that all the input features are independent of each other. We calculate the frequency of each class and then calculate the conditional probability by counting the frequency of that feature within each class.

|  |  |  |
| --- | --- | --- |
| Confusion Matrix | Positive (1) | Negative (0) |
| Positive (1) | (true positive) **108** | (false positive) 8 |
| Negative (0) | (false negative) 1 | (true negative) 7 |

Accuracy: 93.1034% | Precision: 99.08% | Recall: 93.91% | F1: 0.9643

This model is still a pretty good model for the task, but it falls short of the previous logistic regression model. It predicts positive cases incorrectly but cuts the number of false negatives in half. Based on the F1 score alone, this model is inferior. It is important to note, however, that this model is a lot less compute-intensive to train and is a much more streamlined process than the logistic regression, which takes 500+ iterations to train in most cases.

# Problem 4

This section introduces us to a new method of preprocessing data and