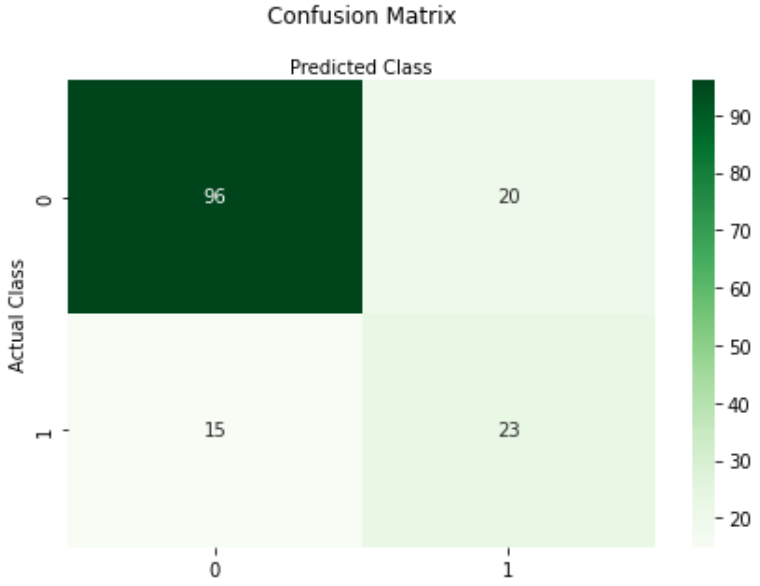
GitHub Code Repository: https://github.com/pharmon0/Harmon_ECGR5105

Problem 1)

Using the diabetes dataset, build a logistic regression binary classifier for positive diabetes. Please use 80% and 20% split between training and evaluation (test). Make sure to perform proper scaling and standardization before your training. Report your results, including accuracy, precision, and recall. At the end, plot the confusion matrix representing your binary classifier.

The Linear Regression was performed with min/max and normal scaling. It was found that the min/max scaling performed better, so that method was carried through the problem. The Regression was run, and the results were as follows:

```
Model Accuracy:  77.273%
Model Precision: 53.488%
Model Recall:    60.526%
```



Problem 2)

Repeat problem 1, and this time use K-fold cross-validation for your training and validation. Perform the training two times for K=5 and K=10. Analyze and compare your average accuracy against problem 1.

This time, it was found that the normal distribution scaling gave better results, so this scaling was carried through the problem. The results can be seen below. The 5-fold cross-validation model performed better on accuracy than the train-test split model, but the 10-fold model was worse than both.

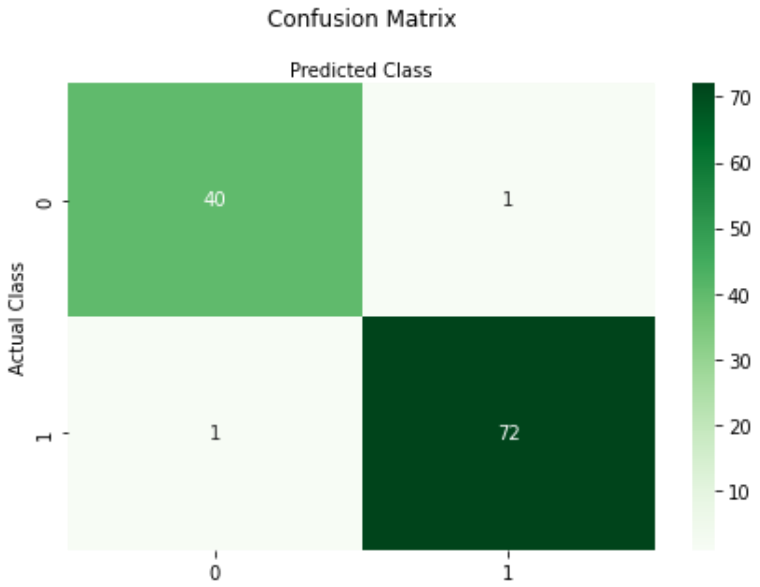
```
K= 5 | Accuracy: 77.734% (2.167%)
K=10 | Accuracy: 76.946% (3.075%)
```

Problem 3)

1. Use the cancer dataset to build a logistic regression model to classify the type of cancer (Malignant vs. benign). First, create a logistic regression that takes all 30 input features for classification. Please use 80% and 20% split between training and evaluation (test). Make sure to perform proper scaling and standardization before your training. Report your results, including accuracy, precision, and recall. At the end, plot the confusion matrix representing your binary classifier.

It was found that the normal distribution scaling performed best for input scaling, and as such, it was used throughout the rest of this problem. The results of the validation can be seen below.

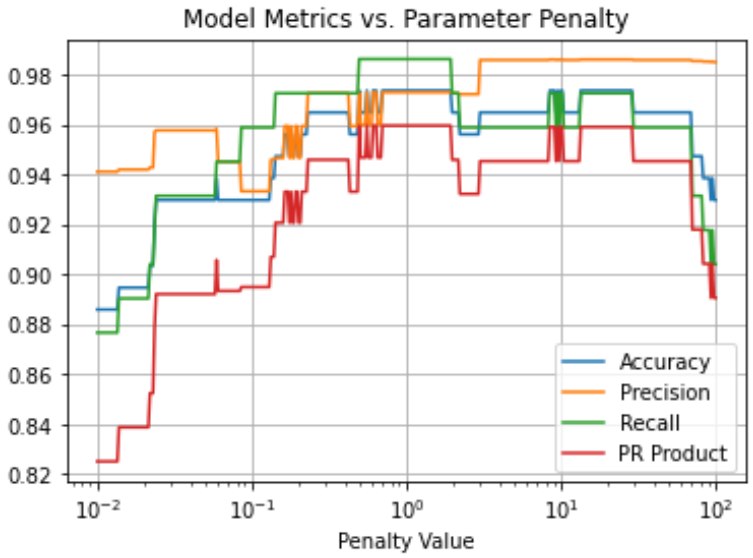
```
Model Accuracy:  98.246%
Model Precision: 98.630%
Model Recall:    98.630%
```



2. How about adding weight penalty here, considering the number of parameters. Add the weight penalty and repeat the training and report the results.

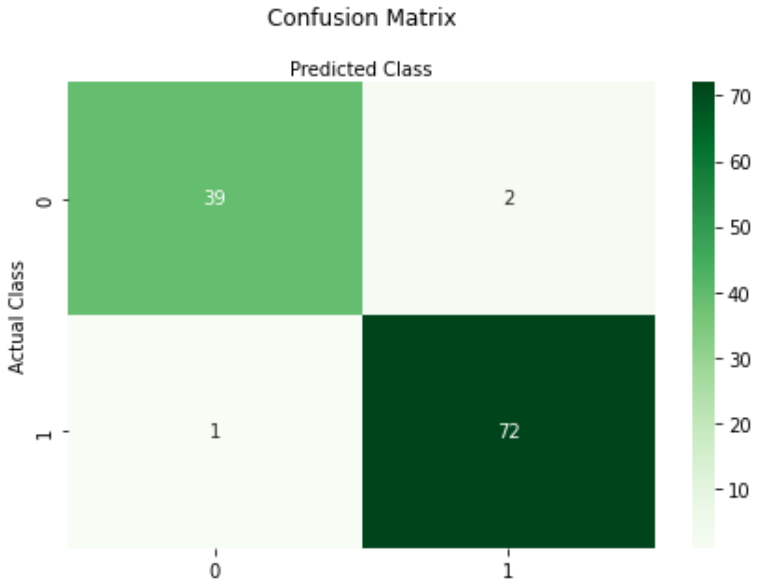
Afterwards, a reevaluation was performed over a wide range of weight penalty values with a log distribution. The results of which can be seen in the plot below.

GitHub Code Repository: https://github.com/pharmon0/Harmon_ECGR5105



From this, lambda = 1 was chosen as the best value, and the resultant data was as follows:

```
Model Accuracy:  97.368%
Model Precision: 97.297%
Model Recall:    98.630%
```



It seems that for this particular method, there is no benefit to the parameter penalty.

GitHub Code Repository: https://github.com/pharmon0/Harmon_ECGR5105

Problem 4)

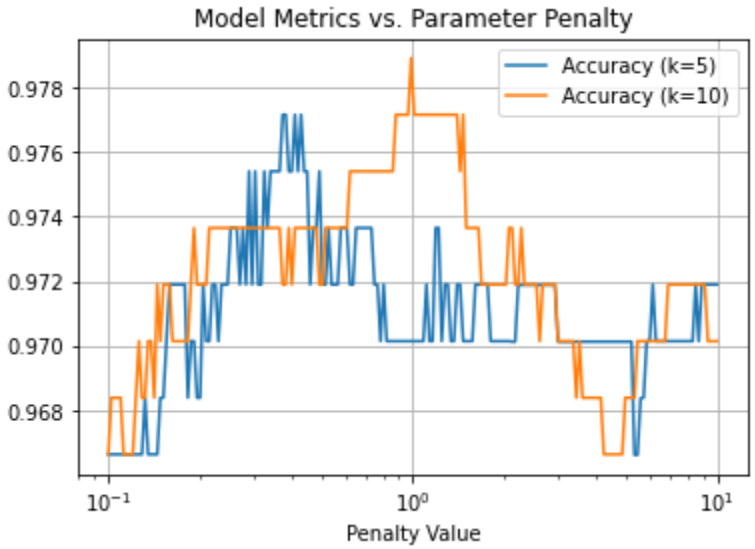
1. Repeat problem 3, and this time use K-fold cross-validation for your training and validation. Perform the training two times for K=5 and K=10. Analyze and compare your average accuracy against problem 3.

It was found that the normal distribution produced better results in this case, so this scaling was used for the rest of this problem. The results of the K-folds methods on this dataset can be seen below. Without any parameter penalization, this model is clearly worse than the train-test split model, regardless of the K-value.

```
K=5 | Accuracy: 97.539% (1.290%)
K=10 | Accuracy: 97.716% (1.122%)
```

2. How about adding weight penalty here, considering the number of parameters. Add the weight penalty and repeat the training and report the average accuracy.

A variety of lambda values with logarithmic distribution were applied to both the 5-Fold and the 10-Fold cross-validation models, and the resultant accuracy graph can be seen below.



From this, approximate-best lambda values were chosen as 0.35 for k=5, and 1 for k=10. This resulted in the following data, leading to the conclusion that the parameter penalty did, in fact, improve the K-folds cross-validation model, if only slightly.

```
K= 5 | Accuracy: 97.541% (0.653%)
K=10 | Accuracy: 97.892% (1.312%)
```


GitHub Code Repository: https://github.com/pharmon0/Harmon_ECGR5105