Grayson Bass
ID Number: 801074960
Homework 2
Github: https://github.com/gbass2/IntroToML

Problem 1:

The metrics for the model are below:

Accuracy: 0.7532467532467533
Precision: 0.6491228070175439
Recall: 0.6727272727272727

The data was split into 80% training and 20% validation. A random state of 42 was used to randomize the splitting of the data so that the data is split more evenly between the training and validation. This helps prevent the training set or validation set from having more of one outcome. The LogisticRegression function apart of the sklearn library was used to create the logistic regression classifier. Approximately 25% of the predicted data was incorrect predictions while approximately 75% were correct predictions. A precision of 0.649 means that when we predict that a patient has diabetes, we are correct 64.9% of the time. A recall of 0.672 means that we predicted 67.2% of the actual positive cases as true positives and 32.8% as false negatives. Figure 1 shows the confusion matrix using the linear regression model. Since the dataset has many more outcomes of 0 than 1, there are more true negative predictions on the validation set. The figures are in the source code file "".

Problem 2:

Problem 2 was completed the same way as problem 1 except a Gaussian Naïve Bayes classifier from sklearn was used instead of the logistic regression classifier. The metrics for the model are below:

Accuracy: 0.7662337662337663
Precision: 0.6610169491525424
Recall: 0.7090909090909091

The accuracy was slightly improved when using the Naïve classifier. The recall and precision also increased compared to the logistic classification model. This means the model was able to predict slightly more true positives and slightly less false negatives. This can be seen in the confusion matrix in figure 2.

Problem 3:

K-Fold cross validation was used with a logistic regression model and achieved by using the built-in libraries apart of sklearn. Both 5 and 10 folds were used. The metrics for both 5 and 10 folds are below:

Accuracy (K=5): 0.7682454800101859
Precision (K=5): 0.7155969191270859
Recall (K=5): 0.5726989972369619
Accuracy (K=10): 0.7707621326042379
Precision (K=10): 0.7173983781918565
Recall (K=10): 0.5800508774379743

Using 10 folds compared to 5 folds did improve the model, but the difference between 5 and 10 folds with the logistic regression model is less than 1%. When comparing the results with problem 1 using the 10 folds, there was a little over a 2.15 % increase in accuracy, 7.8% increase in the precision, and a 9.36% decrease in the recall. Using K-Fold cross validation helped improve the metrics compared to just using logistic regression.

Problem 4:

K-Fold cross validation was used with a Gaussian Naïve Bayes model and achieved by using the built-in libraries apart of sklearn. Both 5 and 10 folds were used. The metrics for both 5 and 10 folds are below:

```
Accuracy (K=5):  0.7539258127493421
Precision (K=5):  0.6645294767870302
Recall (K=5):  0.6011292482940126
Accuracy (K=10):  0.7512303485987697
Precision (K=10):  0.6537815101446814
Recall (K=10):  0.5938748312619281
```

There is not a significant improvement when using K-Fold cross validation when compared to just using the Naïve Bayes Classifier prediction. This is due to kK-Fold cross validation being meaningless to Gaussian Naïve Bayes. Each iteration will be a separate model so if 10 folds are used then 10 independent models are created. This happens because the features for Naïve Bayes are independent of one another. So, it does not make sense to use K-Fold cross validation with Gaussian Naïve Bayes.