Grayson Bass
ID Number: 801074960
Homework 3
Github: https://github.com/gbass2/IntroToML

Problem 1:

For problem 1, the dataset was imported from the sklearn library which can be found in section In[15]. The code for the logistic regression model was completed the same way as the previous homework. Before the data was split, standardization scaling was used. This can be found in section In[16] of the source code. The data was split into 80% training and 20% validation. A random state of 0 was used for both splitting the data and the model. The model was fit using the training data and the predictions were made using the testing data. Accuracy, precision, and recall was recorded for the model. The code for the model and the metrics can be found in section In[17]. The accuracy, precision, and recall are high when using the 30 features and is in the upper 90%. In section In[18], figure 1 shows a heat map was made using from the confusion and out of 114 predictions only 2 were classified as false positives and 2 as false negatives. You can train the logistic regression model using all 30 features but feature extraction can be used to improve the model.

Problem 2:

For problem 2, Principal Component Analysis was used as a form of feature extraction. This was imported using the sklearn library. A logistic regression model was used with PCA. The number of components that the 30 features were to be extracted to was set and the PCA was fit to the data in section In[19]. The PCA data was then split into training and validation. Different number of components were selected, and the results were measured and compared to find the optimal number. The metrics for the model was calculated over the 30 different number of components possible and was plotted. Arrays were used to store the values at each iteration. Figure 2 shows the accuracy vs the number of iterations with the number of iterations being equal to the number of components used. Figures 3, and 4 show the precision and recall vs the iterations. Using a different random_state will vary the most optimal number of components. Looking at all 3 graphs, the optimal number of components that will achieve the highest accuracy is 1, 7, 11, 12, 14, and 15. This is where the accuracy, precision, and recall are the highest. All the code for problem 2 and the plots can be found in section In[19]

Problem 3:

For problem 3, Linear Discriminant Analysis was used as a form for feature extraction. This was imported from the sklearn library. The number of components that the features was extracted to was 1. The Number of components could not be larger than 1 – number of classes. Since the data has a binary Outcome, the number of components is 1. The features were then fit and transformed to the LDA model. This can be seen in section In[20]. In section In[21], the LDA data is split into the training and validation sets. The LDA model was then fit to the training data. The LDA model uses a Naïve Bayes model for the classification. Accuracy, precision, and recall was recorded. A heat map was made from a confusion and was plotted in figure 5 and is in section In[22]. Comparing the confusion matrix to problem 1, the LDA model did slightly better. The false positives decreased from 2 to 0 but the false positives did not decrease and remained at 2.

Problem 4:

Problem 4 was completed in the same way as problem 3. Instead of the Naïve Bayes model, a logistic regression model was used instead. Section In[23] is the code for the feature extraction and like problem 3, only 1 component was used. Section In[24] shows LDA extracted data split into training and validation sets, and the regression model being fitted to the training data. The accuracy, precision, and recall was recorded and figure 6 shows a heat map of the confusion matrix. Comparing confusion matrix to problem 1 and 2, the number of false negatives compared to problem 1 decreased by 1 but was still 1 higher than problem 2. The logistic regression model did slightly worse than the Naïve Bayes model for LDA.