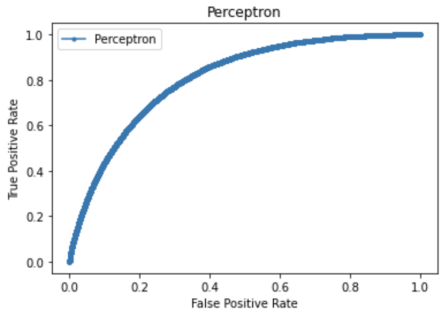
Question 1

1. To prepare the data (for all questions), I normalized all numerical data (BMI, General Health, Mental Health, Physical Health, Age Bracket, and Income Bracket) through by standardization (Z-score normalization), change Biological Sex to be represented by 0s and 1s instead of 1s and 2s (just to match the other categorical variables), and one-hot encoded the remaining categorical variables (Education Bracket and Zodiac). I then split the data into train and test sets with a 70/30 split. I then fit a Perceptron model with all the predictor variables as the inputs and "Diabetes" as the output. After running a Perceptron, I ran the predictions on the test set to calculate my accuracy. Finally, I plotted the graph to show the ROC curves and the AUC.
2. I prepared the data the way I did (for all questions) because for some of the models used in this homework (ex. logistic regression), it's important to work with normalized data. Regarding specifically what I did for the Perceptron model, I did this to see how well all the predictor variables together did at accurately predicting diabetes using Perceptron.
3. I found that when using all predictor variables, the Perceptron model has an accuracy of 86.093% and an AUC of 0.80945.
4. The AUC for the model using all the predictor variables is quite high at 0.80945 (so it is a well-performing model).

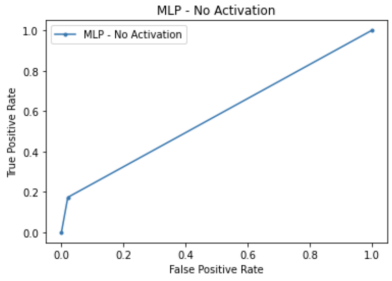



```
The accuracy for Perceptron is 86.093%
AUC = 0.80945
```

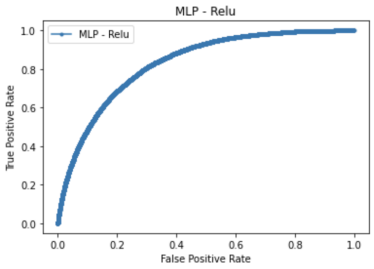
Question 2

1. I first split the data into train and test sets with a 70/30 split. I then fit an MLPClassifier model with all the predictor variables as the inputs and "Diabetes" as the output, specifying the hidden layer. Furthermore, I varied the type of activation functions using relu, sigmoid, identity, and tanh. I also tried using no activation function. After running the models, I ran the predictions on the test set to calculate my accuracies. Finally, I plotted the graphs to show the ROC curves and the AUCs.
2. I did this since the question asked for a feedforward neural network so I used a MLPClassifier to satisfy this. Furthermore, since I was also asked to change the activation function, I changed between relu, sigmoid, identity, and tanh to see the differences between them. Finally, I used no activation function to also see how this would affect my model.

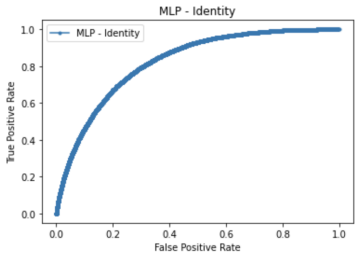
3. All of the accuracies and AUCs are listed below with the visuals. As we can see, sigmoid has the highest accuracy at 86.617% and the second highest AUC at 0.83111 (tanh has a negligibly higher AUC at 0.83140). When checking to see the different layers (using sigmoid as the activation function), I saw that having 3 hidden layers gives us the highest AUC at 0.83178 when using the sigmoid activation function and only varing the number of hidden layers.
4. Compared to Perceptron, I see that using a feedforward neural network improves the AUC slightly (but not by much). Furthermore, we can also see that the choice of activation function doesn't really matter for AUC in networks that aren't that deep (e.g. 1 layer).



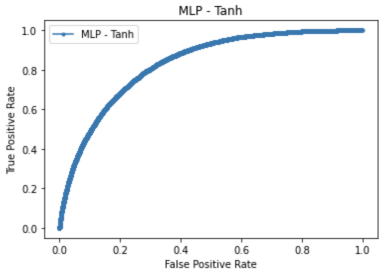
The accuracy for MLP is 86.559%
AUC = 0.57638



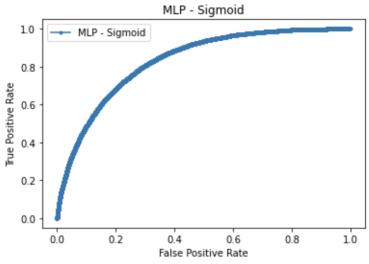
The accuracy for MLP is 86.559%
AUC = 0.83061



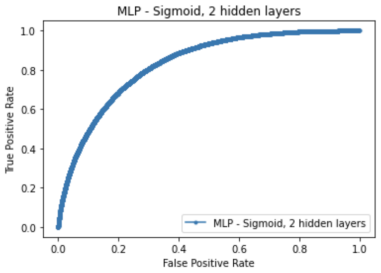
The accuracy for MLP is 86.321%
AUC = 0.82290



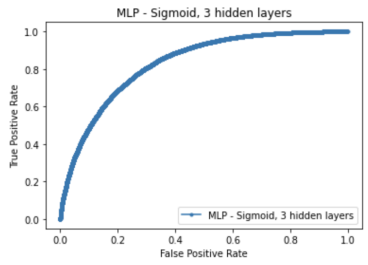
The accuracy for MLP is 86.597%
AUC = 0.83140



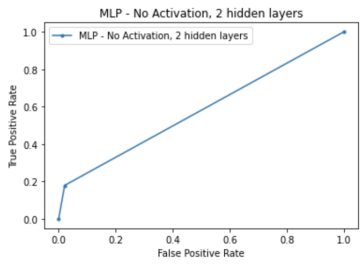
The accuracy for MLP is 86.617%
AUC = 0.83111



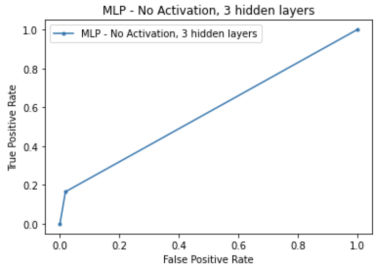
The accuracy for MLP is 86.593%
AUC = 0.83159



The accuracy for MLP is 86.592%
AUC = 0.83178



The accuracy for MLP is 86.596%
AUC = 0.57900



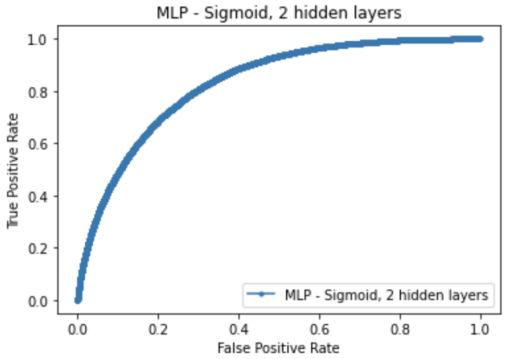
The accuracy for MLP is 86.614%
AUC = 0.57321

Question 3

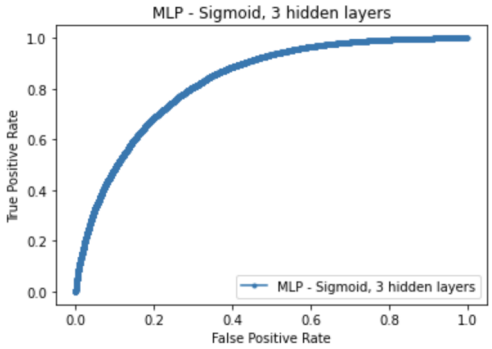
1. I first split the data into train and test sets with a 70/30 split. I then fit an MLPClassifier model with all the predictor variables as the inputs and "Diabetes" as the output. I used the Sigmoid activation function for the MLPClassifier and varied the number of hidden

layers. After running the models, I ran the predictions on the test set to calculate my accuracies. Finally, I plotted the graphs to show the ROC curves and the AUCs.

2. I did this since a “deep” network has at least two hidden layers so I varied the number of hidden layers by having two and three hidden layers.
3. When there are two hidden layers, we get an accuracy of 86.593% and an AUC of 0.83159. When there are three hidden layers, we get an accuracy of 86.592% and an AUC of 0.83178.
4. Based on the above values, I see that the extra hidden layer results in a slightly lower accuracy but a slightly higher AUC. This means that the extra hidden layer is better at predicting diabetes. Given the nature of this dataset, there is no benefit of using a CNN or RNN for classification because CNN is useful when the data has a spatial or grid-like structure (which this isn’t) and RNN is useful when the data has a sequential or temporal structur (which this also isn’t).




```
The accuracy for MLP is 86.593%
AUC = 0.83159
```




```
The accuracy for MLP is 86.592%
AUC = 0.83178
```

Question 4

1. I first split the data into train and test sets with a 70/30 split. I then fit an MLPRegressor model with all the predictor variables as the inputs and “BMI” as the output. I varied the type of activation functions using relu, sigmoid, identity, and tanh. Finally, I calculated the RMSE using the predictions and the test set.
2. I used MLPRegressor because BMI is a continuous variable and therefore wouldn’t work with MLPClassifier. I varied the activation function to see if/how the model changes if I vary this.
3. I found that using the tanh function generates the lowest RMSE at 0.8195920025755244 and using the identity function generates the highest RMSE at 0.8578617291599534. Additionally, sigmoid and relu give RMSEs of 0.8216374441134272 and 0.8250421663974333 respectively.
4. Other than the identity function, we see that the RMSE values are similar when using different activations. This shows that it is a shallow network.

Question 5

1. I first split the data into train and test sets with a 70/30 split. I then fit an MLPRegressor model with all the predictor variables as the inputs and “BMI” as the output. I used two hidden layers and sigmoid activation for the models, the first model having ten neurons, the second model having fifteen neurons, and the last model having thirty neurons. Finally, I calculated the RMSE using the predictions and the test set.
2. I varied the number of neurons to see if changing this would improve the accuracy of the model which would be shown in a lower RMSE score.
3. I found that the model that uses ten neurons generates an RMSE of 0.8207124566433445, the model that uses fifteen neurons generates an RMSE of 0.8190709457000241, and the model that uses thirty neurons generates an RMSE of 0.817375398379225.
4. Based on these values, we see that increasing the number of neurons in each hidden layer leads to better prediction of BMI.

Extra Credit 1

1. I first split the data into train and test sets with a 70/30 split. For my baseline model, I fit an MLPClassifier model with all the predictor variables as the inputs and “Diabetes” as the output. I put no activation function and had two hidden layers. Then, to create another model using only the most importance features, I did L1-based feature selection using a Lasso object as the estimator, setting the alpha to be 0.001. From there, I then split the data again into train and test splits with a 70/30 split. I then fit an MLPClassifier model with the most important predictor variables as determined by the L1-based feature selection and “Diabetes” as the output. I also put no activation function and had two hidden layers. After running the models, I ran the predictions on the test set to calculate my accuracies.
2. I did this to see how using only the “most importance” predictor variables (so removing predictor variables that had effectively no impact on the accuracy) would affect the accuracy and AUC in comparison to the model that used all the predictor variables available and to find exactly what these predictor variables are.
3. I found that the model that used all the predictor variables had an accuracy of 86.596% and that the model that used only the most important predictor variables had an accuracy of 86.641%. Furthermore, I found that that the most important predictor variables were 'HighBP', 'HighChol', 'BMI', 'Smoker', 'Stroke', 'Myocardial', 'PhysActivity', 'HeavyDrinker', 'GeneralHealth', 'MentalHealth', 'PhysicalHealth', 'HardToClimbStairs', 'BiologicalSex', 'AgeBracket', and 'IncomeBracket'.
4. Since we saw the accuracy when removing certain predictor variables, we know that the model performed better when removing these predictor variables. Furthermore, since we found the most important predictor variables to be the ones listed above, we know that there are in fact predictor variables that have effectively no impact on the accuracy of our models. The predictor variables that have effectively no impact on the accuracy of our models are 'Fruit', 'Vegetables', 'HasHealthcare', 'NotAbleToAffordDoctor', 'Kindergarten', 'Elementary', 'HighSchool', 'GED', 'College', 'Graduate', 'Aries', 'Taurus', 'Gemini', 'Cancer', 'Leo', 'Virgo', 'Libra', 'Scorpio', 'Sagittarius', 'Capricorn', 'Aquarius', and 'Pisces'.

Extra Credit 2

Some pros of using neural networks for classification as opposed to classical classification methods are their ability to model highly non-linear relationships between predictor variables and the outcome variables, handle a large number of input predictor variables, handle missing/noisy data and outliers, and to model complex interactions between predictor variables. Some cons of using neural networks for classification as opposed to classical classification methods are them being computationally expensive, prone to overfitting, and less interpretable. As this dataset has many input predictor variables that vary in type (ex. numerical, categorical, etc.) and relationship to the outcome variable, neural networks are a powerful tool for learning this complex dataset. Despite, this, however, neural networks also have limitations such as them being computationally expensive, having potential for overfitting, and being difficult to interpret.