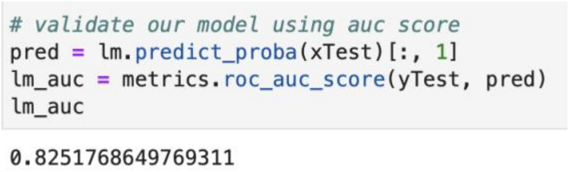
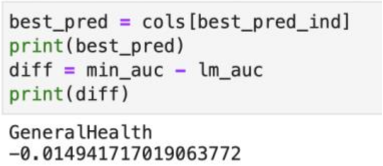
0. Preprocessing
    a. I read in the csv file and extracted the diabetes column out as y and extracted other 21 columns out as predictors X. Since the data is carefully curated according to the spec sheet, I didn't check for NA values.
    b. Since among predictors there are continuous variables like BMI and binary values like HighBP, I normalized the data to make the predictiors in the same scale. Note that when I ran the model on the data with only one predictor dropped for the problem below, this technique was also used.
    c. Then I did the 80-20 train-test split on the dataset with a random_state of 0 to set the train and test dataset. This is to let all model share the same train and test dataset, which is best for comparing model performance.
1. Build a logistic regression model. Doing so: What is the best predictor of diabetes and what is the AUC of this model?
    a. I ran a logistic regression model on all 21 predictors of the train dataset using 'liblinear' solver and balanced class_weight, and I validated the model using the AUC score on the test dataset. To find the best predictor, I ran logistic regression model with each predictor dropped and chose the predictor whose disappearance decreased the AUC the most.
    b. I ran a logistic regression model because this is what the question asked for and it is a good approach on classification problem like this. I used a liblinear solver and balanced class_weight because there is no presumption that whether a class should weigh higher. I calculated the probability that a person is classified as diabetes because it is needed when computing the AUC score as an input. I computed the AUC score with the test data y and the probability of predicted class because it is a common metrics to validate the model of classification problem. I excluded each predictor and ran the linear regression model with remaining predictors to see which model drops the most to find the best predictors because it is the most straightforward thing to do to get the best predictor according to the hint.
    c. The result of the model with all predictors is 0.8252, as shown below.



```
# validate our model using auc score
pred = lm.predict_proba(xTest)[:, 1]
lm_auc = metrics.roc_auc_score(yTest, pred)
lm_auc
```

```
0.8251768649769311
```

And the best predictor is GeneralHealth because when I excluded each predictor and train and test the model with other predictors it gives the highest decrease of 0.0150 AUC score of the models from overall predictors, as shown below.



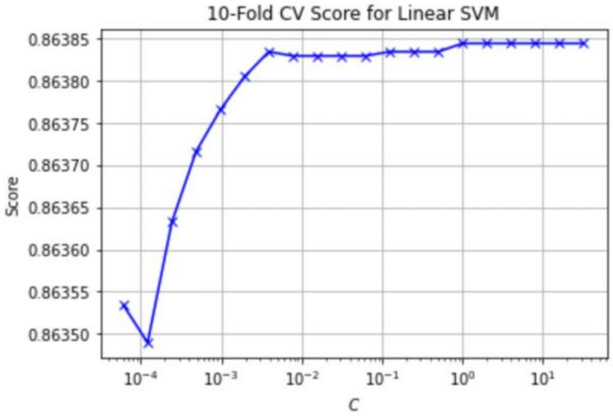
```
best_pred = cols[best_pred_ind]
print(best_pred)
diff = min_auc - lm_auc
print(diff)
```

```
GeneralHealth
-0.014941717019063772
```

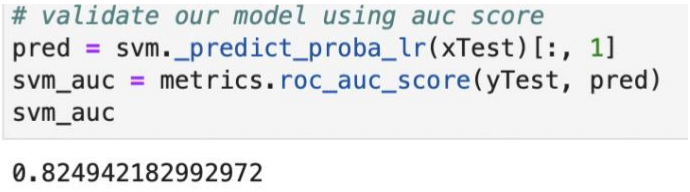
   d. I think this model gives us an overall good performance when classifying whether a person has diabetes based on all the predictors because it has a rather high AUC score (between 0.8-0.9, and 1 is the highest possible value). And it seems reasonable that general health is the best predictor of predicting diabetes according to our common sense.

2. Build a SVM. Doing so: What is the best predictor of diabetes and what is the AUC of this model?
   a. I chose to run a soft margin linear SVM and before I ran a SVM model, I first did hyperparameter tuning of the slack variable and draw a graph. I ran a 10-fold cross validation to find the slack variable with rather the highest score according to the graph. Then, I ran a soft margin linear SVM model with the tuned slack variable on the train dataset, and I validated the model using the AUC score on the test dataset. To find the best predictor, I ran soft margin linear SVM model with the tuned slack variable with each predictor dropped and chose the predictor whose disappearance decreased the AUC the most.
   b. The reason I chose to run a soft margin linear SVM is that I couldn’t determine if such a dataset with that many of dimensions is strictly linearly separable, hence instead of hard margin linear SVM, the soft margin linear SVM is introduced because it could allow some misclassified edge data. Also, a linear model would be much faster. I did the hyperparameter tuning because I want to find the best hyperparameter C such that could give the model the best performance, and I used 10-fold cross validation because cross validation was a good approach to find the best hyperparameter and to make sure the model does not overfit. I calculated the probability that a person is classified as diabetes because it is needed when computing the AUC score as an input. I computed the AUC score because it is a common metrics to validate the model of the classification problem. I excluded each predictor and ran the model with others to see which model drops the most to find the best predictors because it is the most straightforward thing to do according to the hint.

c. The result of the slack variable I used is 1.0 according to the graph.

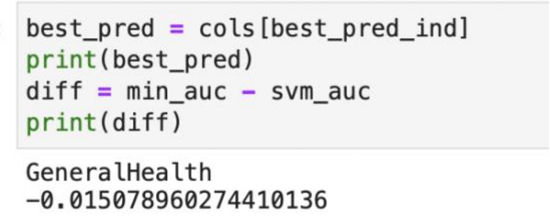


The result of the model with slack variable 1.0 with all predictors is 0.8249, as shown below.


```
# validate our model using auc score
pred = svm._predict_proba_lr(xTest)[:, 1]
svm_auc = metrics.roc_auc_score(yTest, pred)
svm_auc
```

```
0.824942182992972
```

And the best predictor is GeneralHealth because when I dropped each predictor and train and test the model with other predictors it gives the highest decrease of 0.0151 AUC score of the model from overall predictors as shown below.


```
best_pred = cols[best_pred_ind]
print(best_pred)
diff = min_auc - svm_auc
print(diff)
```

```
GeneralHealth
-0.015078960274410136
```

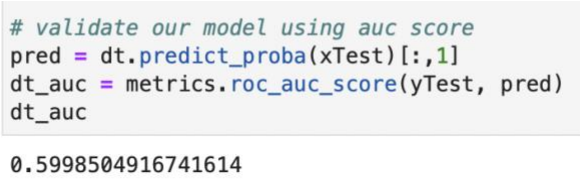
d. I think this model gives us an overall good performance when classifying whether a person has diabetes based on all the predictors because it has a rather high AUC score (between 0.8-0.9, where 1 is the highest possible value). And it seems reasonable that general health is the best predictor of predicting diabetes.

3. Use a single, individual decision tree. Doing so: What is the best predictor of diabetes and what is the AUC of this model?

   a. I ran a single, individual decision tree on the train dataset using ‘entropy’ criterion, and I validated the model using the AUC score on the test dataset. To find the best predictor, I ran the single, individual decision tree with each predictor dropped and chose the predictor whose disappearance decreased the AUC the most.

b. I calculated the probability that a person is classified as diabetes because it is needed when computing the AUC score as an input. I computed the AUC score because it is a common metrics to validate the model. I dropped each predictor and ran the model with others to find the best predictors because it is the most straightforward thing to do according to the hint.

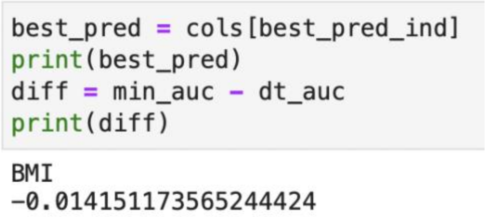
c. The result of the model with all predictors is 0.5999 as shown below.



```
# validate our model using auc score
pred = dt.predict_proba(xTest)[:,1]
dt_auc = metrics.roc_auc_score(yTest, pred)
dt_auc
```

```
0.5998504916741614
```

And the best predictor is BMI because when I dropped each predictors and train and test the model with other predictors it gives the highest decrease of 0.0142 AUC score of the model from overall predictors as shown below.



```
best_pred = cols[best_pred_ind]
print(best_pred)
diff = min_auc - dt_auc
print(diff)
```

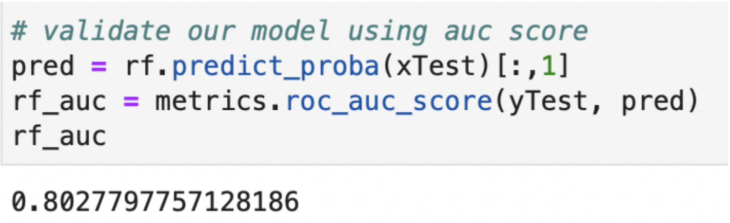
```
BMI
-0.014151173565244424
```

d. I think this model gives us a rather weak performance compared to other models (between 0.5-0.6, where 0.5 is the lowest possible value). It makes sense because a single decision tree is easy to overfit to the training data set and it has tendency to be weak learners, which causes weak performance on the testing data set. Therefore, we need the random forest or adaboost to bootstrap the single decision trees and get strong learner model and a higher AUC score. Note that it seems reasonable that BMI is the best predictor of predicting diabetes, because we always hear that diabetes is somewhat related with weight, intuitively.

4. Build a random forest model. Doing so: What is the best predictor of diabetes and what is the AUC of this model?
   a. I ran a random forest model on the train dataset using ‘gini’ criterion, a max_feature of 0.5, and 100 estimators, and I validated the model using the AUC score on the test dataset. To find the best predictor, I ran a random forest model with each predictor dropped and chose the predictor whose disappearance decreased the AUC the most.
   b. I ran a random forest model because this is a common good approach for a classification problem and it is a strong learner compared to a single decision

tree. I used max_feature of 0.5 because with higher max_feature there might be overfitting issues. Also, the random forest will choose a random 0.5 of the features when creating single decision tree in the bootstrapping process and so I don’t need to worry that there are important features that are not covered. I chose 100 estimators because it is enough estimators for creating a strong learning model for this problem according to the AUC score and it won’t take the model too long to run. I calculated the probability that a person is classified as diabetes from the model because it is needed when computing the AUC score as an input. I computed the AUC score because it is a common metrics to validate the model of classification problem. I dropped each predictor and ran the model with others to find the best predictors because it is the most straightforward thing to do according to the hint.

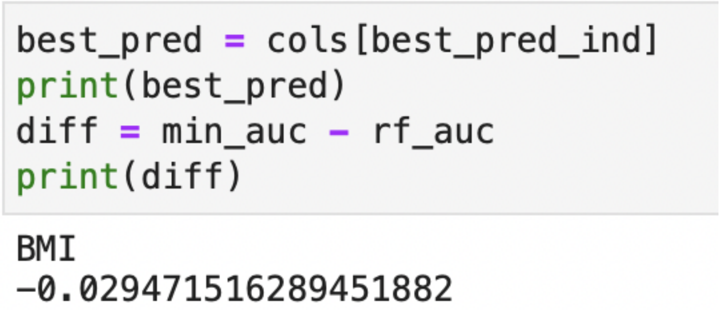
c. The result of the model with all predictors is 0.8028, as shown below.



```
# validate our model using auc score
pred = rf.predict_proba(xTest)[:,1]
rf_auc = metrics.roc_auc_score(yTest, pred)
rf_auc
```

```
0.8027797757128186
```

And the best predictor is BMI because when I dropped each predictor and train and test the model with other predictors it gives the highest decrease of 0.0295 AUC score of the model from overall predictors, as shown below.



```
best_pred = cols[best_pred_ind]
print(best_pred)
diff = min_auc - rf_auc
print(diff)
```

```
BMI
-0.029471516289451882
```

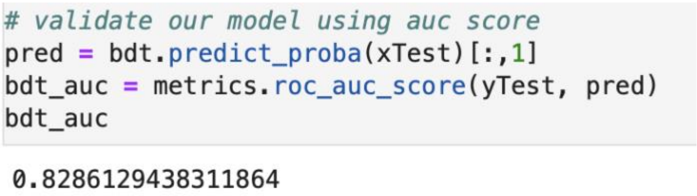
d. I think this model gives us an overall good performance when classifying whether a person has diabetes based on all the predictors because it has a rather high AUC score (between 0.8-0.9, where 1 is the highest possible value). And it seems reasonable that BMI is the best predictor of predicting diabetes.

5. Build a model using adaBoost. Doing so: What is the best predictor of diabetes and what is the AUC of this model?

    a. I ran an adaBoost on the train dataset using a max depth of 1 of the tree bootstrapping, ‘gini’ criterion, a learning rate of 1, and 100 estimators, and I validated the model using the AUC score on the test dataset. To find the best predictor, I ran an adaBoost with each predictor dropped and chose the predictor whose disappearance decreased the AUC the most.

b. I ran an adaBoost model because it is a strong learner compared to a single decision tree. I used max_depth of 1 of the decision tree classifier inside the adaboost because in adaboost we are supposed to create a forest of stumps. I use a learning rate of 1 because a higher one might result in overlooking the minimum value of the cost function, and a lower one might take the model too long to train. I calculated the probability that a person is classified as diabetes because it is needed when computing the AUC score as an input. I computed the AUC score because it is a common metrics to validate the model of classification problem. I dropped each predictor and ran the model with others to find the best predictors because it is the most straightforward thing to do according to the hint.

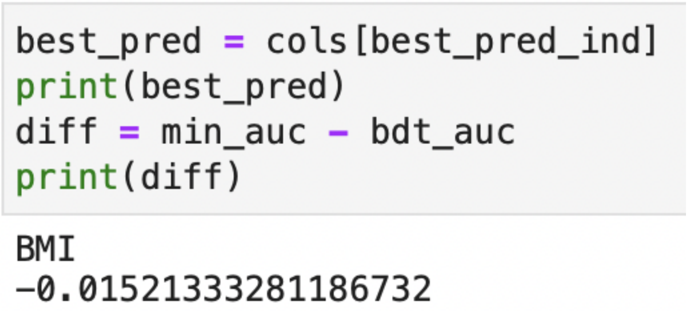
c. The result of the model with all predictors is 0.8286, as shown below.



```
# validate our model using auc score
pred = bdt.predict_proba(xTest)[:,1]
bdt_auc = metrics.roc_auc_score(yTest, pred)
bdt_auc
```

```
0.8286129438311864
```

And the best predictor is BMI because when I dropped each predictor and train and test the model with other predictors it gives the highest decrease of 0.0152 from overall predictors, as shown below.



```
best_pred = cols[best_pred_ind]
print(best_pred)
diff = min_auc - bdt_auc
print(diff)
```

```
BMI
-0.01521333281186732
```

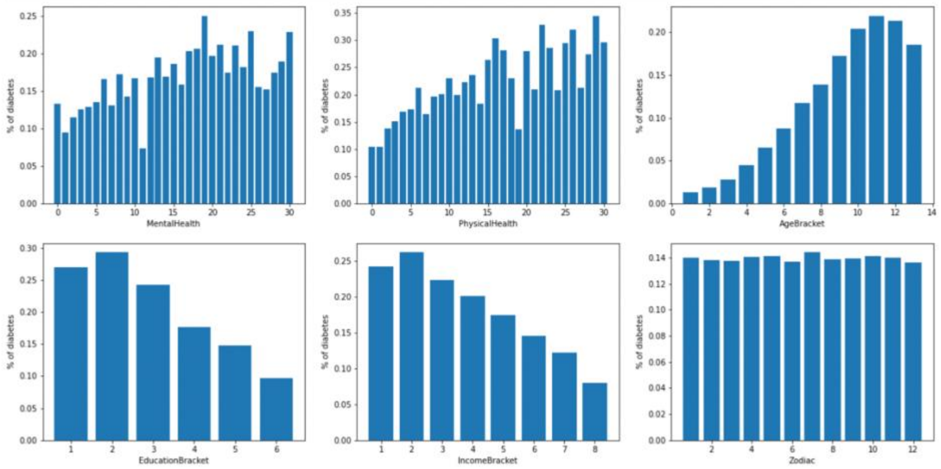
d. I think this model gives us an overall good performance when classifying whether a person has diabetes based on all the predictors because it has a rather high AUC score (between 0.8-0.9, where 1 is the highest possible value). And it seems reasonable that BMI is the best predictor of predicting diabetes.

EC:

1. Comparing the result performance of all 5 models, I found that adaboost has the highest AUC score of approximate 0.8286 on the overall 21 predictors. Because they are all

trained and tested with the 80-20 datasets, I deduced that adaboost is the best model to predict diabetes in this dataset.

2. In this question I try to find two interesting insights:
   a. I find the two variables with the highest correlation score.
      i. I drew the correlation matrix of each variables and maintain a global variable to keep track of the max absolute of the matrix.
      ii. The result shows that GeneralHealth and PhysicalHealth have a correlation of 0.52
      iii. This result is very interesting because it is not at all intuitive to me. I assume that not only the correlation between GeneralHealth and PhysicalHealth should be higher but also the highest correlation should also be higher because there are many variables that are potential highly correlated. Take, Fruit and Vegetables as an example, a person who likes eating fruit should also like eating vegetables in my assumption. It turns out such assumptions might be wrong because of the relatively low correlation score computed from data. Sometimes, data could give us counterintuitive insights.
   b. I find the distribution of diabetes in each of the bracket variables.
      i. For each bracket variable 'GeneralHealth', 'MentalHealth', 'PhysicalHealth', 'AgeBracket', 'EducationBracket', 'IncomeBracket', 'Zodiac', I calculated the percentage of reported diabetes (number of diabetes/number of records fallen in this bracket). This percentage is calculated to standardize the diabetes cases in each bracket. (if not there might be a bracket with fewer people reported and another bracket with more people reported)
      ii. Then I draw the histograms of all these bracket variables. The result is shown below.



      iii. As shown in the picture, the diabetes cases percentage is right skewed distributed, indicating that people reporting more bad mental health and physical health days in the last 30 days tend to have higher percentage appearance of diabetes. It is also right skewed in Ages, indicating that older people tend to have higher percentage of diabetes cases among them. However, the graph is left distributed for EducationBracket and IncomeBracket. It means that people with lower education and lower income tend to have higher percentage getting diabetes. Note that the diabetes is uniformly distributed across all the zodiac, indicating that there is not a perceivable tendency regarding this variable. This is quite interesting because we don’t see these tendencies in the five models presented above and they are not the best predictors. However, they do give us some informative insights like tendencies across all brackets variables.