Qingyu Zhang / Andy

N-number: N19903322

Prof. Pascal Wallisch

Fundamentals of Machine Learning

HW3 Report: Diabetes

Question 1:

1）What was done:

Tested the dataset and found no nan values. The total number of features was 21 and the outcome variable was diabetes status (first column). Split the dataset into training set and test set (80%/20%). Standardize the feature data for the training and test sets. Fit a baseline logistic regression model with the training set data (using all 21 features). Calculate the AUC scores on the test set and draw ROC plot to evaluate the baseline model.

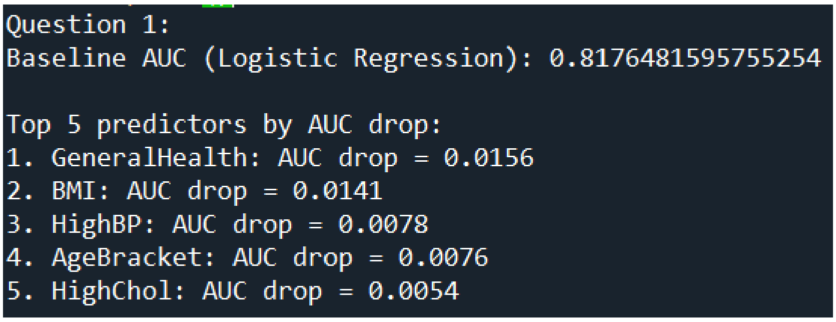
To find the best predictor, delete individual features accordingly and fit the model to record how much the deletion of the feature decreases the AUC compared to the AUC of the baseline model using all features. The top five features that decrease the AUC the most after deletion and the specific values of the decrease are shown.

2）Why this was done:

Not much needs to be said about fitting models and such, it's all standard operation. The main thing that needs to be mentioned is the method of finding the best predictor. I use one of the straightforward methods mentioned in hint. By comparing the difference between the AUC of the model fitted after removing a certain feature and the AUC of the model before removing it, we can get a clear idea of how much this feature contributes in the predictive power of the model. If removing a feature decreases the predictive power of the model by a lot, it means that this feature is a good predictor. On the other hand, the AUC score is one of the best references to measure the performance of a classifier model, and it is also a required scoring metric for questions. So, we use AUC drop to find the best predictor.

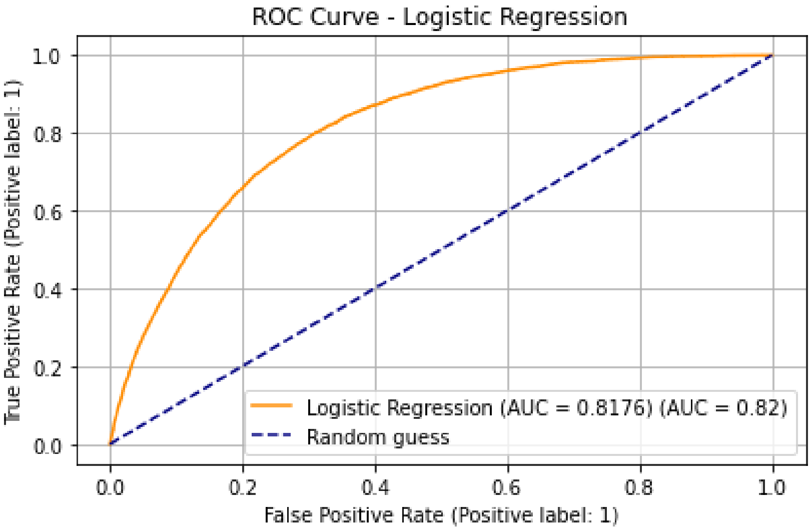
3) Findings:

The AUC of the logistic regression baseline model fitted using all 21 features was approximately **0.8176.** The ROC image is shown. The top five best predictors sorted according to AUC drops are shown.



```
Question 1:
Baseline AUC (Logistic Regression): 0.8176481595755254

Top 5 predictors by AUC drop:
1. GeneralHealth: AUC drop = 0.0156
2. BMI: AUC drop = 0.0141
3. HighBP: AUC drop = 0.0078
4. AgeBracket: AUC drop = 0.0076
5. HighChol: AUC drop = 0.0054
```



4) Interpretation:

As shown in the output, **the best predictor is General Health** (if remove it, AUC of the model will drop the most (drop about 0.0156)). Nonetheless, removing the second-best predictor (Body Mass Index (BMI)) drops the model's AUC by 0.0141. These two values are about the same, meaning that General Health is only a little bit better than BMI. BMI is a good predictor as well.

The **AUC of the baseline logistic regression model is 0.8176**.

Question 2:

1) What was done:

Basically, it's pretty much the same as in Question 1. After all, this time the five questions just ask us to do the same thing with five different models. In this question I use the LinearSVC() function to fit the linear support vector machine classifier to predict diabetes and find the best predictor for this model in the same way as before.

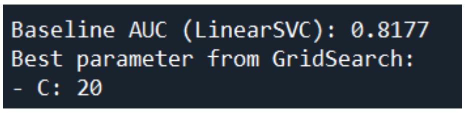
The special part of the code for this question is the hyperparameter tuning part. The LinearSVC() function requires the specification of the penalty strength / slack variable C. I use the GridSearchCV() function to conveniently tune the parameter on the training set (evaluated by the criterion roc_auc). I set the hyperparameter C to be tuned to many values ranging from 0.01 to 1000. After finding the best hyperparameter C, that value is used for fitting the baseline LinearSVC model and for all models when finding the best predictor.

2) Why it was done:

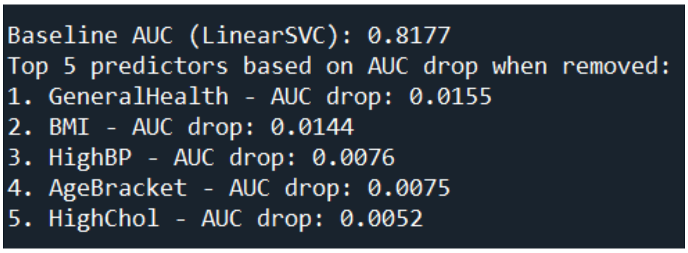
The use of AUC drops to find the best predictor was explained in Question 1 and the same is done in this Question 2. The rest of the same will not be repeated. Focus on the hyperparameter tuning part. For many models, including SVM, we should not just pick a random value as a hyperparameter setting. The GridSearchCV() function makes it easy to use different hyperparameter settings in the training set for cross-validation (split into training and validation sets, and test the effect of different hyperparameters in the validation set). After finding the relatively optimal hyperparameters, we can then proceed to model training, evaluation, find the best predator and other steps.

3) Findings:

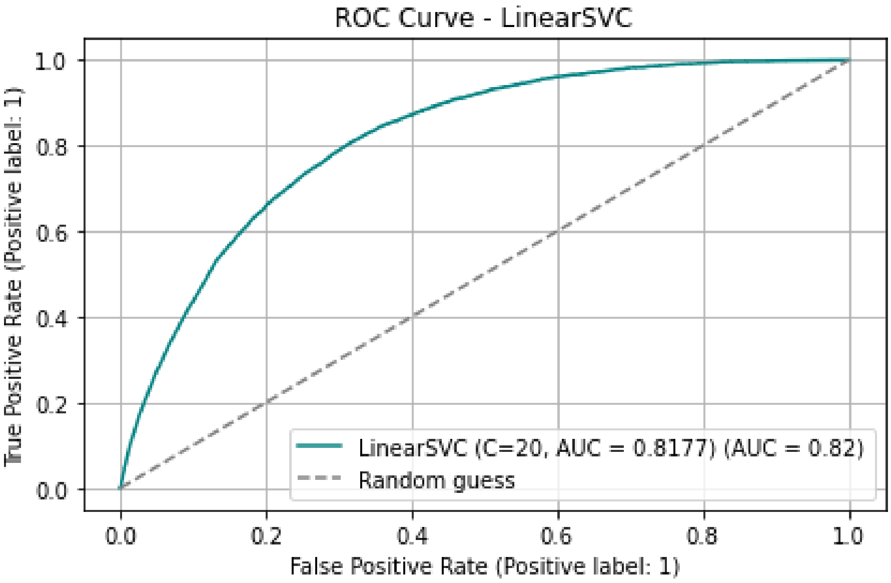
The relative best hyperparameter C is 20. The AUC of the Linear SVC baseline model fitted using all 21 features was approximately **0.8177**. The ROC image is shown. The top five best predictors sorted according to AUC drops are shown.



```
Baseline AUC (LinearSVC): 0.8177
Best parameter from GridSearch:
- C: 20
```



```
Baseline AUC (LinearSVC): 0.8177
Top 5 predictors based on AUC drop when removed:
1. GeneralHealth - AUC drop: 0.0155
2. BMI - AUC drop: 0.0144
3. HighBP - AUC drop: 0.0076
4. AgeBracket - AUC drop: 0.0075
5. HighChol - AUC drop: 0.0052
```



4) Interpretation:

As shown in the output, **the best predictor is General Health** (if remove it, AUC of the model will drop the most (drop about 0.0155)). Nonetheless, removing the second-best predictor (Body Mass Index (BMI)) drops the model's AUC by 0.0144. These two values are about the same, meaning that General Health is only a little bit better than BMI. BMI is a good predictor as well.

The **AUC of the baseline Linear SVC model is 0.8177**.

Question 3:

1) What was done:

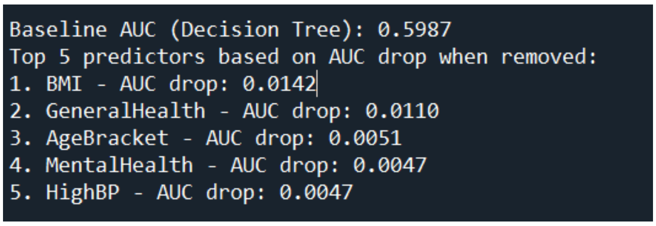
The 3rd question uses a single decision tree, so no hyperparameter tuning is needed. Use Gini impurity. Similar to the previous two questions, fit a baseline decision tree using all features and compute the AUC. Remove features one by one and record the corresponding AUC drops.

2) Why it was done:

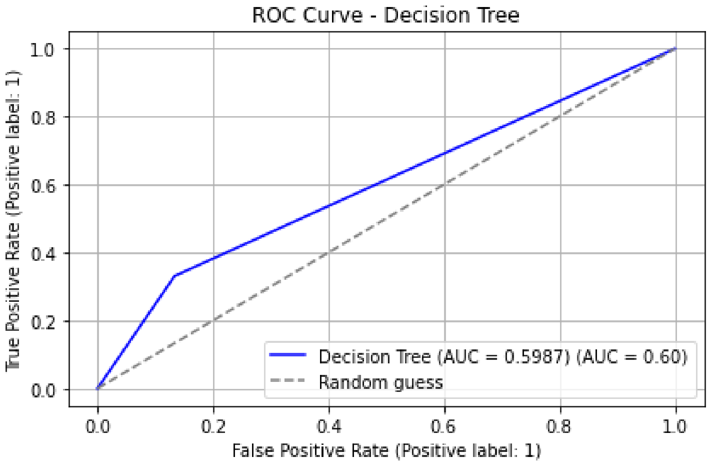
Other things that are similar to the previous two questions will not be repeated here. Focus on why Gini was chosen over entropy. The lab talks about how both decision tree predictions using Gini and entropy perform about the same, but the decision tree fit using Gini is faster and more efficient. So, use Gini. The reason for not standardizing the features is that models based on decision trees ignore scaling. The subsequent Random Forest and AdaBoost also do not require standardization.

3) Findings:

The AUC of the single, individual Decision Tree baseline model fitted using all 21 features was approximately **0.5987**. The ROC image is shown. The top five best predictors sorted according to AUC drops are shown.



```
Baseline AUC (Decision Tree): 0.5987
Top 5 predictors based on AUC drop when removed:
1. BMI - AUC drop: 0.0142
2. GeneralHealth - AUC drop: 0.0110
3. AgeBracket - AUC drop: 0.0051
4. MentalHealth - AUC drop: 0.0047
5. HighBP - AUC drop: 0.0047
```



4) Interpretation:

As shown in the output, **<u>the best predictor is Body Mass Index (BMI)</u>** (if remove it, AUC of the model will drop the most (drop about 0.0142)). As we have seen, unlike the previous two models, BMI became the best predictor when using a single decision tree model. But that's okay. General Health, which was the best predictor in the first two models, is the second-best predictor (AUC drops 0.0110 when removing it) in the single decision tree, just a little worse than BMI.

The **<u>AUC of the baseline single Decision Tree model is 0.5987</u>**. As we see in the ROC plot, the single decision tree is significantly less predictive than the previous logistic regression and linear support vector machine classifier.

Question 4:

1) What was done:

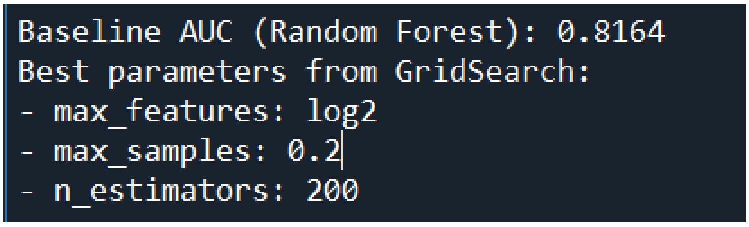
The 4th question requires the use of a random forest model. The code steps for fitting the baseline model and finding the best predictor remain similar to the previous questions. So, the repetition is skipped here. What makes the random forest model different from the previous questions is the hyperparameter tuning part. I used GridSearchCV() to tune three hyperparameters: n_estimators, max_features, and max_samples, and used roc_auc as the hyperparameter tuning evaluation metric.

2) Why it was done:

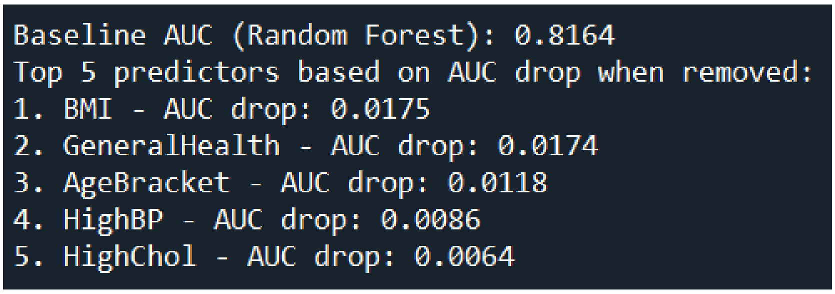
Repetitions such as explanations of model fitting and the method of finding the best predictor are skipped here (since the rationale for the five questions on this section is exactly the same). The reason for tuning the three hyperparameters n_estimators, max_features, and max_samples is that they affect the predictive and generalization abilities of the random forest the most. Using too many decision trees, too large proportion of features per node, and too large proportion of samples per tree may cause the random forest to overfit the training set, which leads to lower generalization ability on the test set. So, it is reasonable to tune them.

3) Findings:

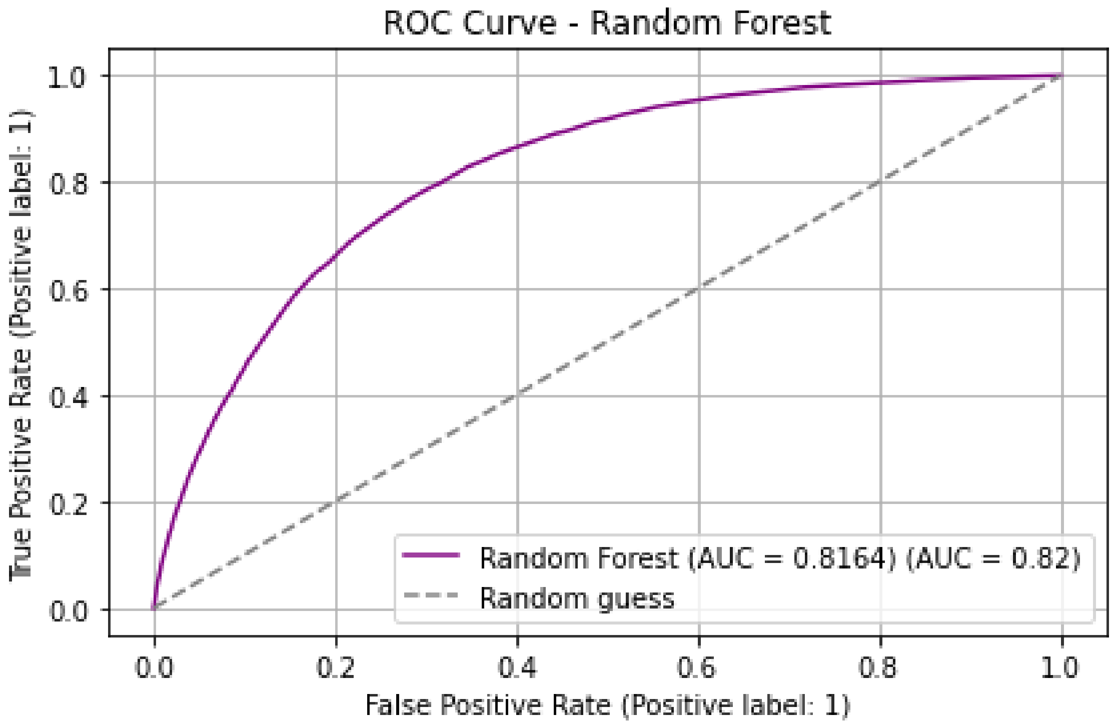
The relative best hyperparameters are shown in the graph. The AUC of the Random Forest baseline model fitted using all 21 features was approximately **0.8164**. The ROC image is shown. The top five best predictors sorted according to AUC drops are shown.



```
Baseline AUC (Random Forest): 0.8164
Best parameters from GridSearch:
- max_features: log2
- max_samples: 0.2
- n_estimators: 200
```



```
Baseline AUC (Random Forest): 0.8164
Top 5 predictors based on AUC drop when removed:
1. BMI - AUC drop: 0.0175
2. GeneralHealth - AUC drop: 0.0174
3. AgeBracket - AUC drop: 0.0118
4. HighBP - AUC drop: 0.0086
5. HighChol - AUC drop: 0.0064
```



4) Interpretation:

As shown in the output, **the best predictor is Body Mass Index (BMI)** (if remove it, AUC of the model will drop the most (drop about **0.0175**)). However, removing **General Health** (the second-best predictor in the Random Forest model) drops the model's AUC by **0.0174**. Compared to the AUC drop for BMI, the difference between the two is only **0.0001**. So, I think that, at least in the Random Forest model, **BMI and General Health** are nearly indistinguishable from each other in terms of being the two best predictors. The **AUC of the baseline Random Forest model is 0.8164**, which is similar to that of logistic regression and Linear SVC.

Question 5:

1) What was done:

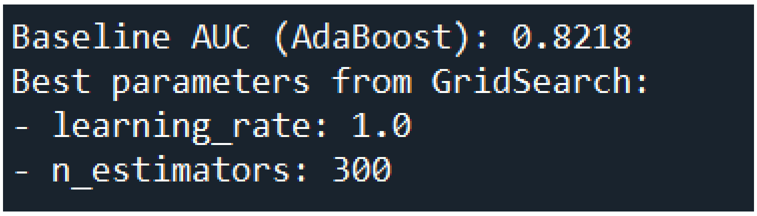
   The 5th question requires the use of the AdaBoost model. The code portion of this question is similar to the random forest in question 4. I'll skip the explanation for repetitive parts. There are two parts that are different. The first is that AdaBoost uses a weak classifier --- decision stump (a decision tree with a maximum depth of 1) as the unit. The second is that the hyperparameter tuning part is different. For AdaBoost, the hyperparameters I tuned were n_estimators and learning_rate. The rest of the steps were the same as for Random Forest.

2) Why it was done:

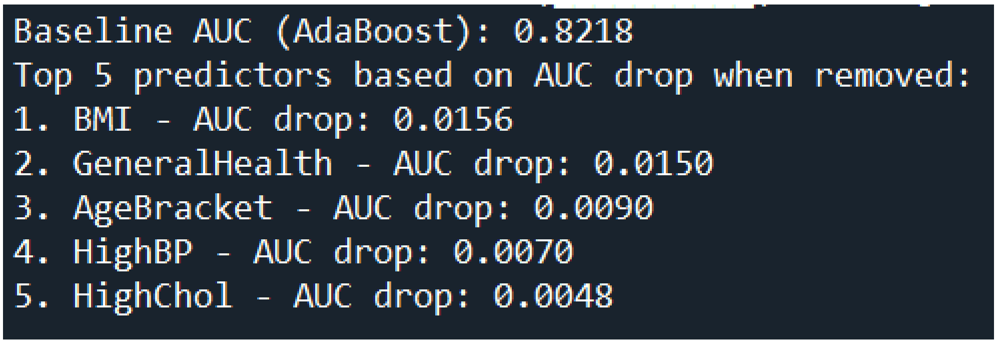
   The reason for tuning n_estimators is that the number of decision stumps affects the model's ability to fit the training set and generalize to the test set. If the number of stumps is too small, there are not enough weak classifiers to synthesize a strong enough classifier, and the Bias of the model will be too high. If the number of stumps is too high, it will overfit the training set, the generalization ability will be low, and the Variance of the model will be too high. The reason for tuning learning_rate is that a too low learning rate will cause the model to fit too slowly and even never reach the optimum; a too high learning rate will cause the model to miss the optimum. Each new stump will overcorrect the errors of the previous stump. Except for the tuning part, all other operations are the same as in Question 4, so I will not explain them again.

3) Findings:

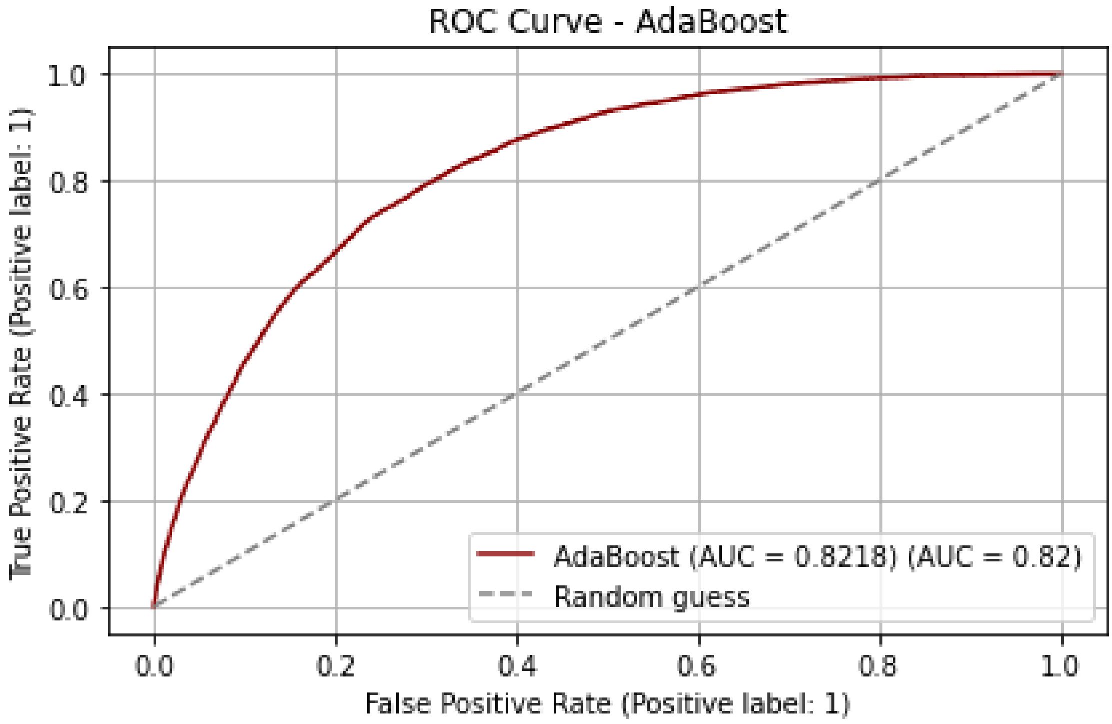
The relative best hyperparameters are shown in the graph. The AUC of the AdaBoost baseline model fitted using all 21 features was approximately **0.8218**. The ROC image is shown. The top five best predictors sorted according to AUC drops are shown.



```
Baseline AUC (AdaBoost): 0.8218
Best parameters from GridSearch:
- learning_rate: 1.0
- n_estimators: 300
```



```
Baseline AUC (AdaBoost): 0.8218
Top 5 predictors based on AUC drop when removed:
1. BMI - AUC drop: 0.0156
2. GeneralHealth - AUC drop: 0.0150
3. AgeBracket - AUC drop: 0.0090
4. HighBP - AUC drop: 0.0070
5. HighChol - AUC drop: 0.0048
```



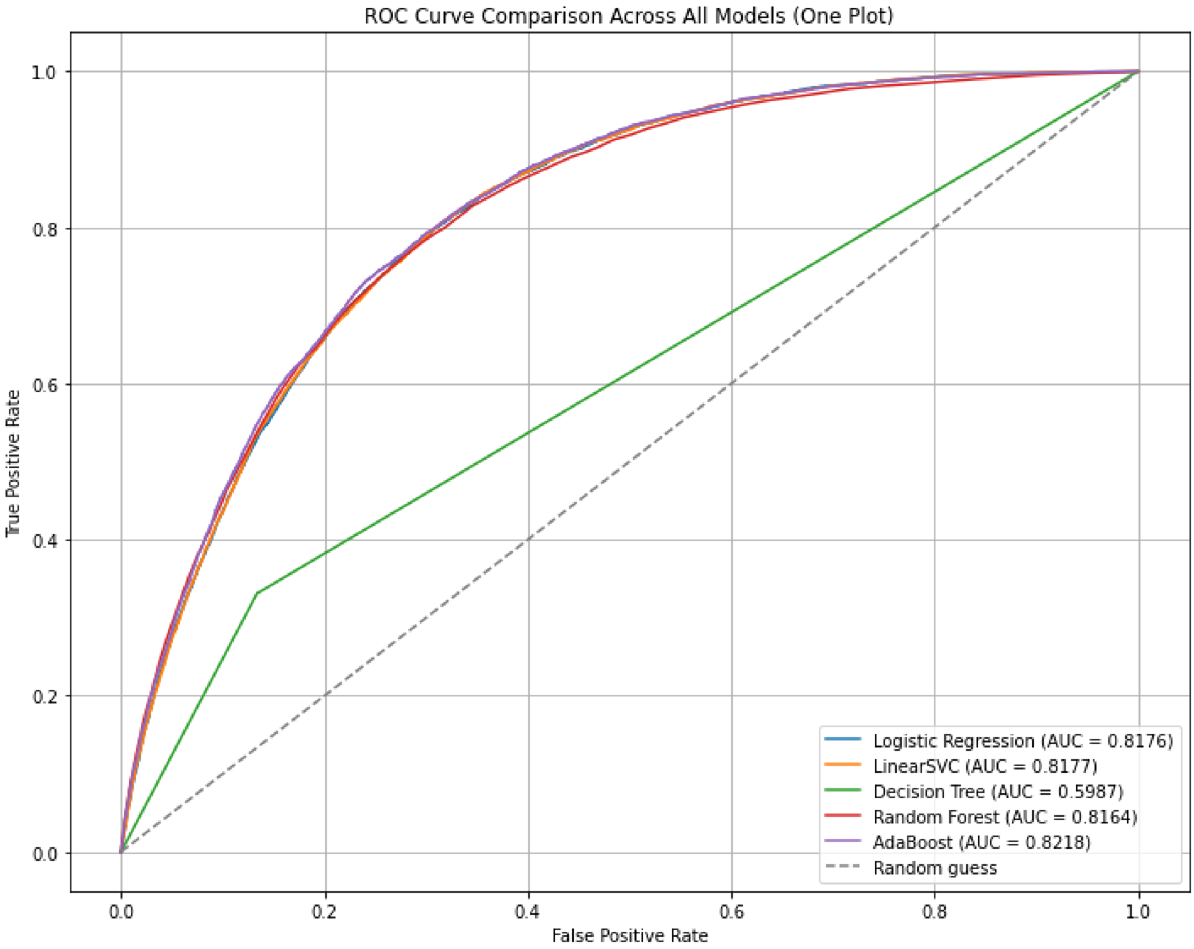
4) Interpretation:

   As shown in the output, **the best predictor is Body Mass Index (BMI)** (if remove it, AUC of the model will drop the most (drop about **0.0156**)). The second-best predictor is General Health (removing it will cause AUC of the model to drop **0.0150**). The two AUC drops are quite close. So, the BMI is only a little bit better than General Health in AdaBoost model.

   The **AUC of the baseline AdaBoost model is 0.8218**, which is similar to that of logistic regression, Linear SVC, and Random Forest.

Extra Credit Problems:

(a) I plotted the ROC curves for the five models on the same graph and labeled them with their respective AUC scores. This makes it easy to see which of them has the best predictive power for diabetes in this dataset.



As we have seen, the predictive power of the other four models is almost indistinguishable from each other, except for the single decision tree, which has much lower predictive power than the other four.

In terms of **AUC values** alone, **AdaBoost is the strongest model** of these five for diabetes prediction (for this dataset).

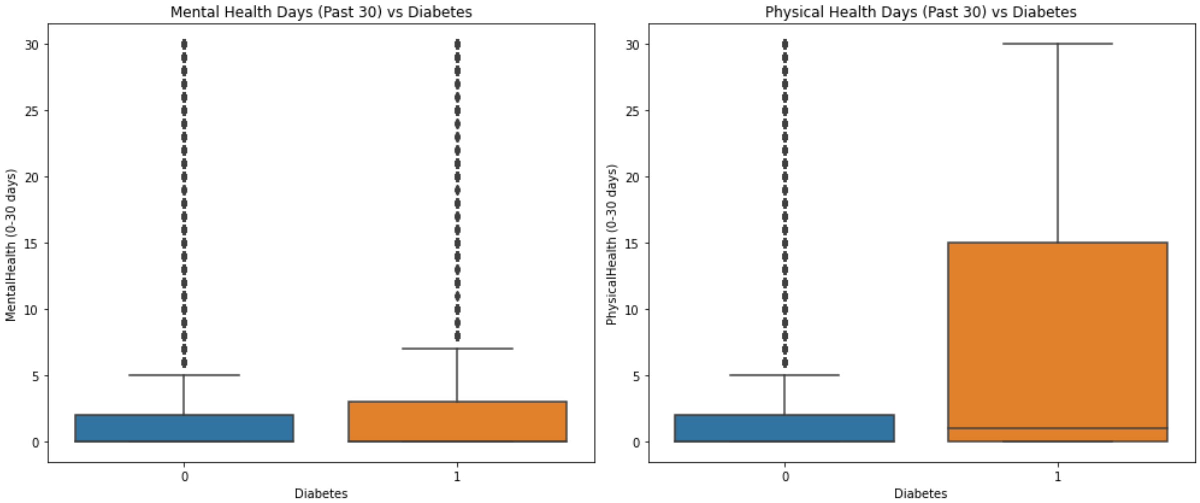
(b) Beyond the questions explored above, one compelling and non-obvious finding from the dataset is the significant difference in both **mental and physical health** between individuals with and without diabetes.

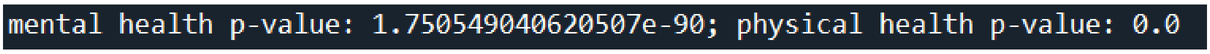
Specifically, diabetic individuals reported experiencing **more mentally and physically unhealthy days** over the past 30 days. This pattern is not immediately intuitive, as diabetes is typically studied through a physiological or lifestyle lens.

We conducted a non-parametric **Mann-Whitney U test** to statistically compare the distributions of self-reported *MentalHealth* and *PhysicalHealth* days between the two groups. The results were highly significant:

- MentalHealth: $p = 1.75 \times 10^{-90}$
- PhysicalHealth: $p < 1e-300$

These results suggest that diabetes may be closely associated with reduced psychological well-being and overall quality of life. This insight points to the importance of considering **mental health support** as part of comprehensive diabetes care and prevention strategies.





```
mental health p-value: 1.750549040620507e-90; physical health p-value: 0.0
```