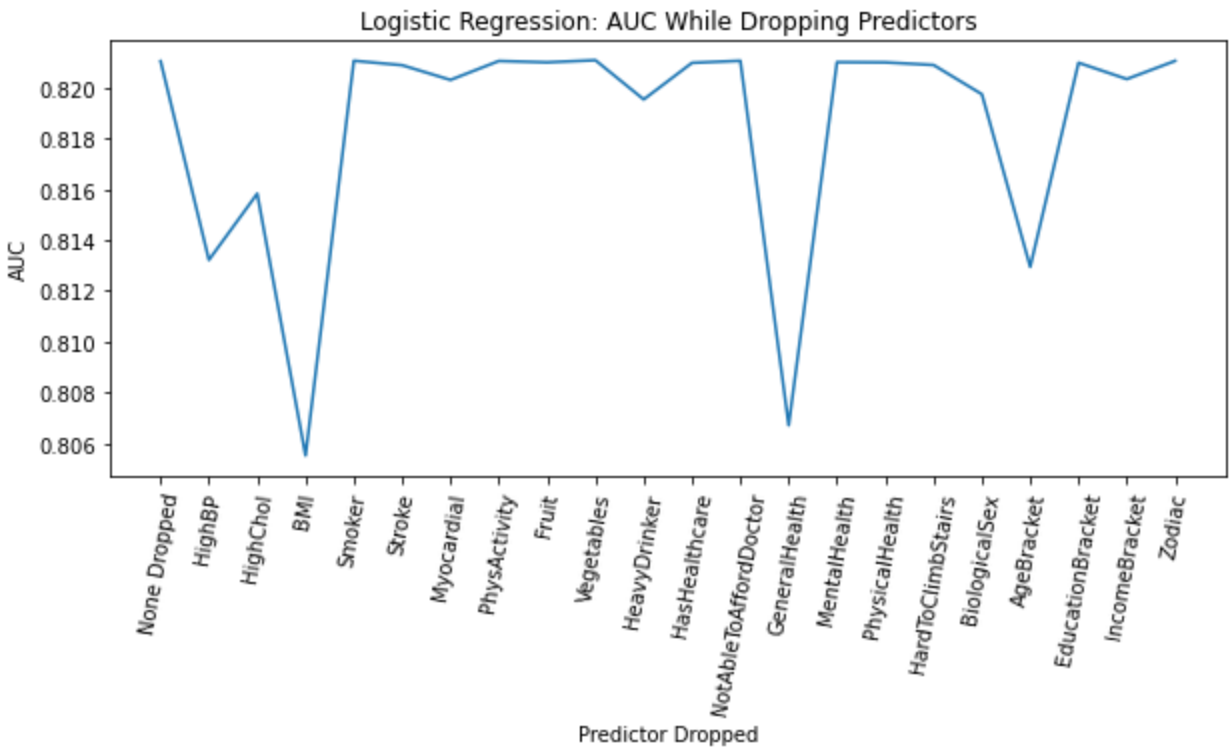
Simone Rittenhouse
Prof. Pascal Wallisch
Into to Machine Learning
6 April 2022

# Intro to Machine Learning: Homework 3

**NOTE:** My code, particularly the GridSearchCV used in questions 4 and 5, takes a long time to run. For your convenience, I've included all relevant code output at the end of this document in case you'd like to reference it without waiting for the code to finish running.

1. **Build a logistic regression model. Doing so: What is the best predictor of diabetes and what is the AUC of this model?**
    a. To find the best predictor using logistic regression, I first split the data into a training and test set, with 30% of the data being used for validation. I then built a logistic regression model using all twenty-one predictors with the outcome variable 'diabetes.' I used the 'liblinear' solver in my model and the class_weight 'balanced.' To find the best predictor, I defined a function that drops predictors one by one from a model by running new models using the twenty remaining predictors each time. For all models, I fit the training data and found the AUC using the test data. I then plotted the AUC across all models and found which one dropped the AUC the most from the full model.
    b. I split my data at the beginning of my code since all the models I used need to be fit to a training set before being evaluated on performance for a test set (to ensure the model doesn't overfit to training data and can generalize). I also didn't normalize or otherwise pre-process the dataset as the majority of the variables were binary and/or categorical. For each logistic regression model, I used the solver because, though it's slower for large datasets than 'sag' or 'saga,' the model still ran fairly quickly. I used a 'balanced' class weight because I found far fewer people diagnosed with diabetes than undiagnosed. I then dropped predictors one-by-one because the 'best' predictor would be the one that is most responsible for the full model's performance. This means that the best predictor would be the one that causes the model to perform the worst after it is excluded.
    c. The AUC of the full logistic regression model was 0.82103. Additionally, BMI dropped the AUC of the model the most when excluded, falling to about 0.806. There was also a sharp decrease when the variables General Health and Age Bracket were dropped as shown in the plot below; however, this decrease was not as large as that of the model excluding BMI.

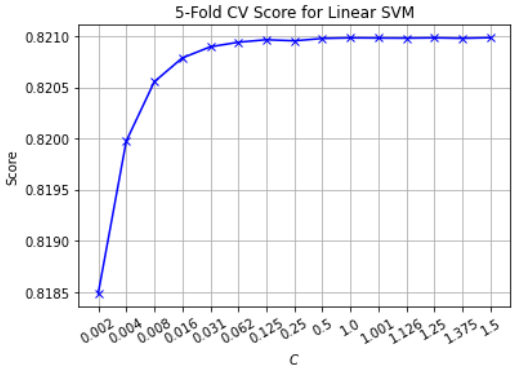


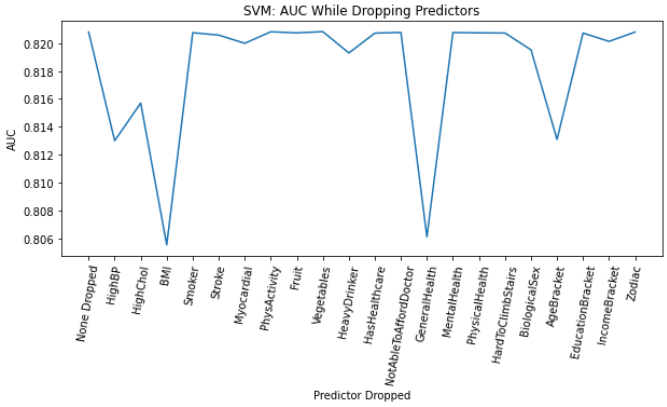

d. The full model's AUC of 0.821 is fairly high, meaning that the model is performing well. However, there is still room for improvement, as a perfect predictor would have an AUC of 1. The model that dropped the predictor BMI and used the remaining twenty predictors had the sharpest decrease in AUC, meaning that BMI is the single variable most responsible for the performance of the full model. Therefore, in the full logistic regression model, BMI is the best predictor of diabetes. General Health and Age Bracket are also strong predictors of diabetes, but not as predictive as BMI.

2. **Build a SVM. Doing so: What is the best predictor of diabetes and what is the AUC of this model?**
   a. To find the best predictor of diabetes in a SVM model, I used the same training and test set from question 1. I then found the optimal slack value to use in my model by finding the 5-fold cross-validation score for AUC on the training set using slack values between 0.002 and 1.5. I built a full linear SVM model using the optimal slack value and all predictors. I found the AUC for the full model using the test data (as in all questions), and then used the same function as in the previous question to drop predictors one by one from the model and plot the change in AUC. Additionally, because LinearSVC has no predict_proba method, I used the decision_function method to find the AUC.
   b. I tested various slack values for the full model because I wanted to ensure that the model was allowing enough misclassification to avoid overfitting on the training

data while optimizing model performance. I tuned the slack variable using cross validation because it allowed me to use the training data to optimize the hyperparameter - meaning I could preserve my test set to evaluate the completed models. The optimal slack value was taken as the first value in which improvement in CV score was below a 0.00001 threshold. This is because the cross validation score improved as C increased but plateaued at higher values, so I took a value that allowed minimal misclassification while preserving a high cross validation score. My logic for finding the best predictor is the same as in question 1. As before, the best predictor should be the one that drops performance the most when excluded.

c. The most optimal slack value tested was 0.125, which can be seen in the plot below as having a high 5-fold cross-validation score while still allowing less misclassification than higher slack values. The AUC of the full SVM model (using C=0.125) was 0.82079. As in question 1, BMI caused the largest decrease in AUC when dropped, followed by General Health and Age Bracket.




d. The full model's AUC is again fairly high (0.82079). The SVM model also had a slightly lower performance than logistic regression in terms of AUC (0.8207 < 0.8210); however, this difference is relatively small. Like the logistic regression model, dropping BMI also caused the largest decrease in AUC, meaning that BMI is the single best predictor of diabetes in the full SVM model. Again, General Health was the second most important predictor, followed by Age Bracket.

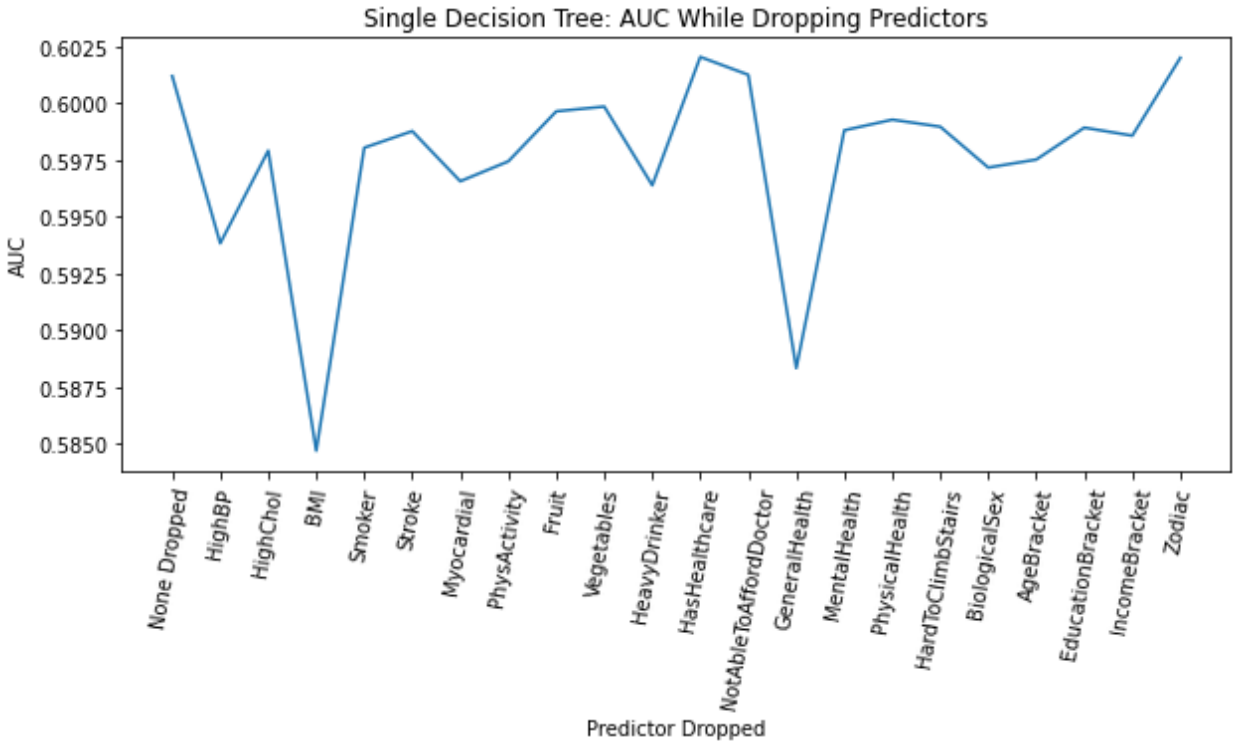
**3. Use a single, individual decision tree. Doing so: What is the best predictor of diabetes and what is the AUC of this model?**

a. To find the best predictor of diabetes using an individual decision tree, I built a full model using all twenty-one predictors and used the test dataset to find AUC. I then re-used the same function as in the previous questions to drop predictors one by one from the model and calculate AUC each iteration. In all of the individual

decision trees, I used sklearn's tree.DecisionTreeClassifier model with random_state = 0 and the Gini Index criterion to measure leaf impurity and determine where the trees should split. I then plotted the AUC across all models and found which predictor dropped the AUC of the full model the most when excluded.

b. I used the Gini Index to measure leaf impurity because entropy requires the calculation of a logarithmic function, and therefore might have a slightly longer runtime. I also set a random_state instead of the default None because I wanted to ensure that my results remained consistent each runtime. I used the same function as in the other questions because I wanted to see which single predictor contributes most positively to the performance of the full model in order to determine the best single predictor of the outcome. Therefore, measuring the largest decrease in AUC caused by the exclusion of a predictor would suggest that that predictor is the best.

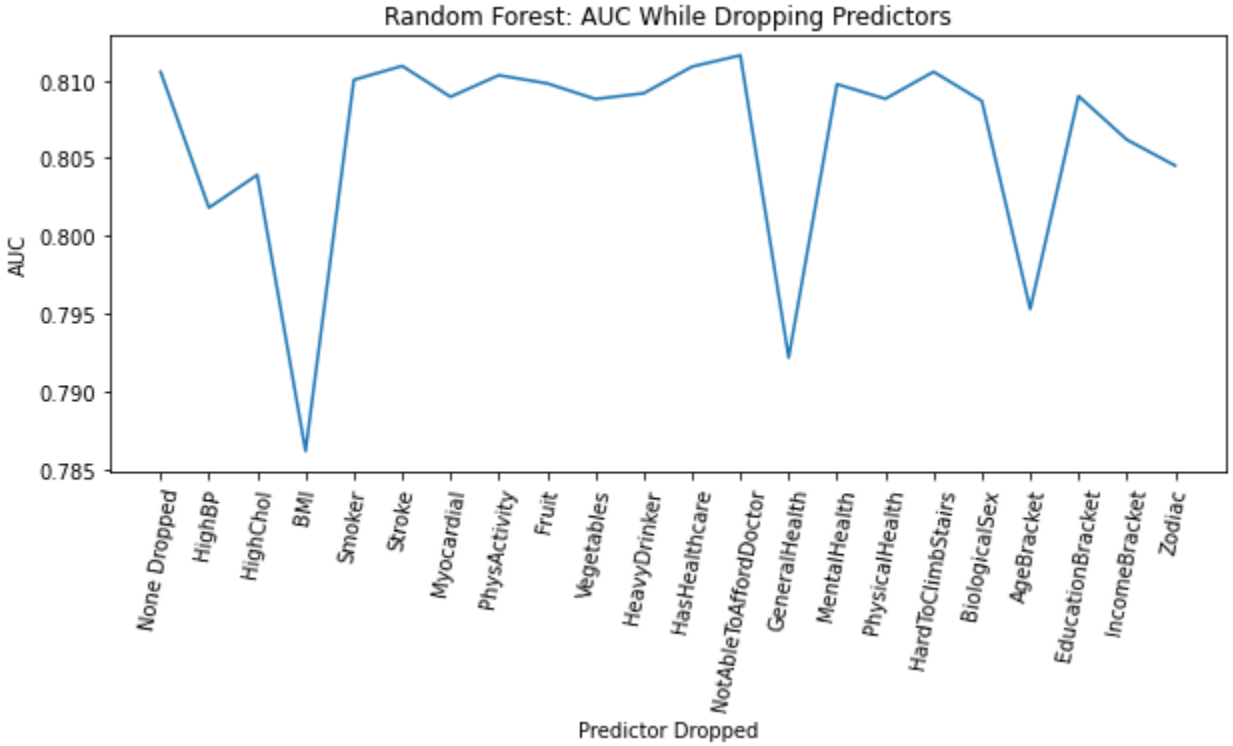
c. The full model had an AUC of 0.60111. Additionally, dropping BMI dropped the AUC of the model the most - resulting in an AUC of about 0.585. General Health caused the second largest decrease in AUC when dropped, leading to an AUC of about 0.588. Interestingly, when I repeated the same process but evaluated accuracy instead of AUC across the models, General Health caused the largest decrease instead of BMI with a full model accuracy of 0.79464. This suggests that both variables contribute largely to overall model performance.



d. Overall, the AUC of a single tree was lower than the AUC obtained by other methods at 0.601. Individual decision trees tend to overfit, so this met my expectation that metrics evaluated on the test set would be lower than in other more robust models. Like the other models, BMI caused the AUC of the model to decrease the most, making it the best predictor evaluated by AUC. I'm a bit suspicious of these results, however, as evaluating by accuracy rather than AUC led to General Health being the best predictor. Still, this makes sense as General Health also dropped the AUC of the model, although not as much as BMI, suggesting it is also an important feature.

4. **Build a random forest model. Doing so: What is the best predictor of diabetes and what is the AUC of this model?**
    a. In order to find the best predictor of diabetes using a random forest model, I used GridSearchCV scoring on AUC to tune the number of estimators, the maximum number of samples to take from the data when training each tree, and the maximum number of features to consider in each tree when deciding where to split. I used the default 5-fold cross validation to optimize the three hyperparameters for the highest average AUC. I then used the optimal values to build a full random forest classifier with 100 estimators. Each estimator used the Gini Index to split, considered 25% of the total features when splitting, and trained on 50% of the data using bootstrapped samples. I found the AUC of the full model and used the same function as before to drop predictors one by one and compute each AUC. I then plotted the AUC across the models and found which predictor caused the AUC to drop the most from the full model.
    b. I used GridSearchCV to tune my hyperparameters because I wanted to test different values for multiple parameters, so a grid search was more efficient than manual cross-validation for each combination of hyperparameters. I used bootstrapped samples from the training data when building each tree and only used at maximum a sample half as large as the full training set to prevent overfitting, as each tree trained on a different random sample. I also tuned max_features to avoid overfitting - which can occur if the trees consider all features in the training set at every split instead of a random sample of them. I used the same function as in the other questions because, again, I wanted to see how the model performed with the exclusion of a single predictor. The predictor that dropped performance the most when excluded would be best.
    c. The AUC of the full model was 0.81044. As in the other questions, BMI dropped model performance the most. General Health had the second largest decrease in AUC (as shown below), followed by Age Bracket; however in this case BMI was the only variable to drop the AUC below 0.790 when excluded. Interestingly, this is the only model in which the Zodiac predictor also visibly dropped the AUC,

suggesting that the inclusion of the Zodiac variable increased the full model's performance.

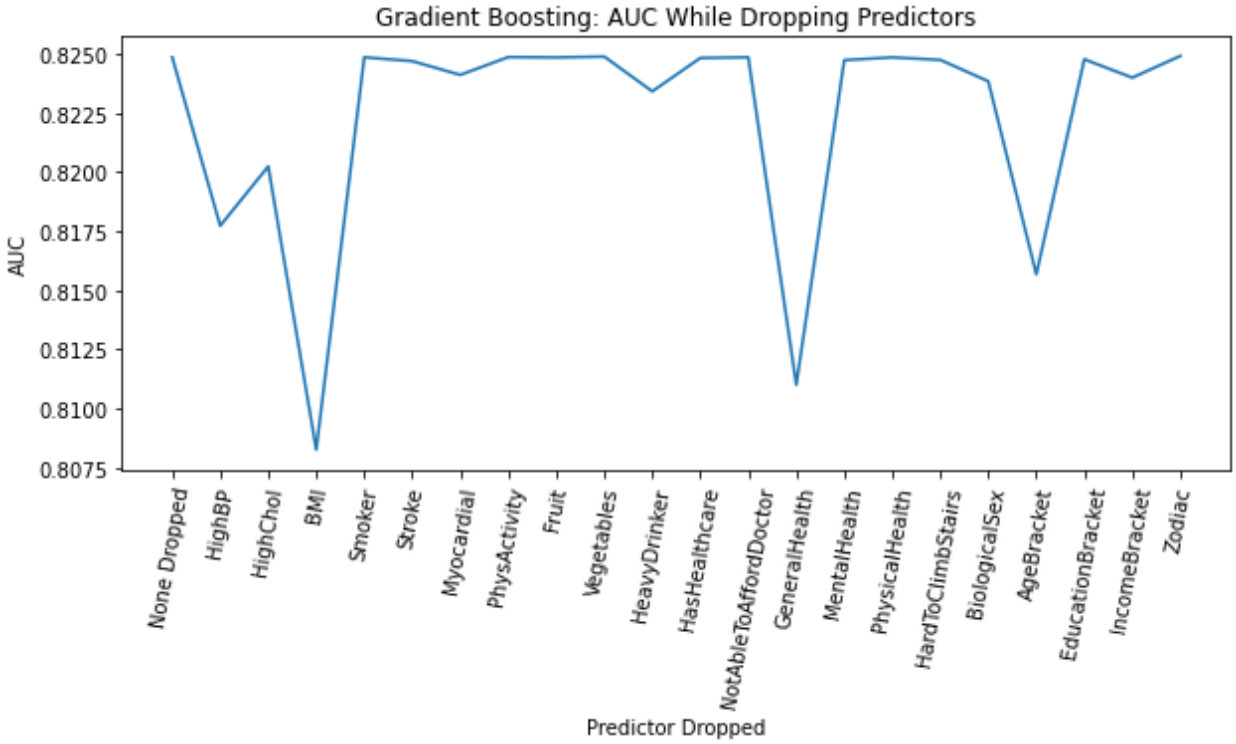



d. The random forest model had an AUC of 0.81044. This is a better performance than the single decision tree (0.81044 > 0.6011) but worse than logistic regression and SVM. I believe model performance could be improved were I to use more estimators in the model; however, this would greatly increase runtime for minimal improvements in performance metrics. Because BMI dropped the AUC the most when excluded from the full model, BMI is again the best predictor of 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. Lastly, to find the best predictor of diabetes using adaBoost, I again used the default 5-fold cross validation score in GridSearchCV to tune the number of estimators and the learning rate - scoring on AUC. I used the optimized hyperparameters from the grid search to build a full adaBoost model, with 500 estimators, a learning rate of 0.1, and a max tree depth of 1 to ensure I was using stumps in the model. I again found the AUC of the full model before using my function to drop predictors one by one from the full model and compute the AUC each iteration. I also plotted the AUC across models to see which predictor dropped AUC the most when excluded.
   b. I used GridSearchCV to test different combinations of hyperparameters while preserving my test data to evaluate the final models. I tested between 100 and 500

estimators as a larger number of estimators than this led to increased runtime without much improvement in model performance. I also tested learning rates between 0.001 and 1, as the optimal learning rate also depends on the number of estimators used in the model. Then, as in the other questions, I dropped predictors and compared AUC because the best predictor would be the one that drops the performance of the model the most when excluded.

c. The AUC of the full model was 0.82485. As with the other models, BMI dropped model performance the most when excluded - leading to an AUC of about 0.8080. General Health dropped model performance the second most, followed by Age Bracket. However, BMI was the only predictor to drop model performance below 0.81, as shown in the figure below.

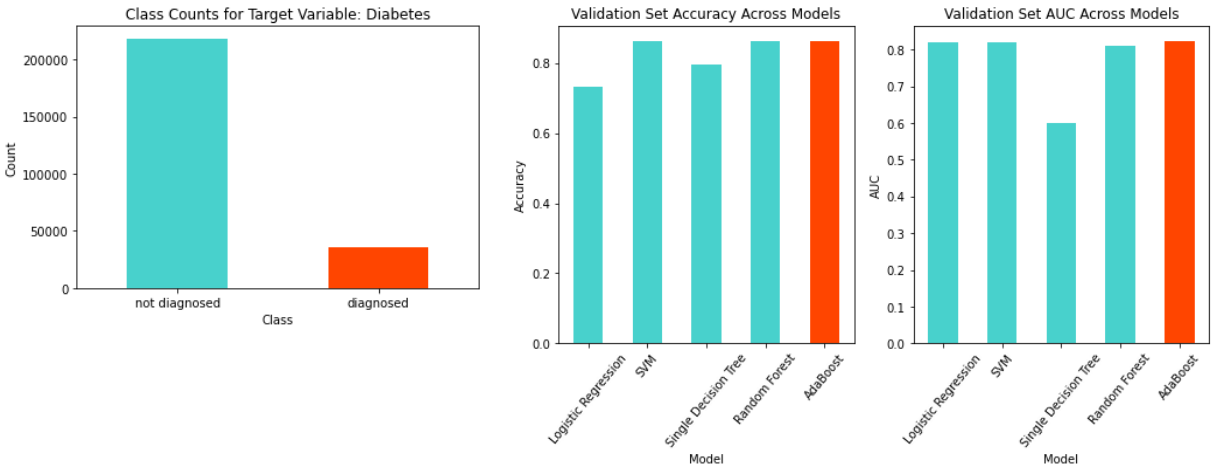



d. The AUC using the adaBoost model was the highest out of all the models (AUC = 0.825). However, as with all the models, there is still room for performance improvement as a perfect AUC score would be 1.0. Additionally, as in the other models, BMI was the best predictor of diabetes since excluding it from the full model led to the largest drop in model performance. General Health and Age Bracket also dropped model performance when removed, although not as significantly, meaning they are also important features.

**Extra credit:**

**A. Which of these 5 models is the best to predict diabetes in this dataset?**

a. To determine which of the five models is the best at predicting diabetes, I compared the AUCs calculated for the full models. I also calculated the accuracy of each of the full models. To compare these values, I created two bar charts showing the accuracy and AUC respectively across the five models. I also plotted the class counts for the outcome variable to examine imbalance - which helped inform which metric I should use to compare models. Lastly, I found the model with the highest accuracy and the model with the highest AUC.
b. I plotted the class counts for the outcome variable because, as discussed in lecture, accuracy is a poor metric if the outcome variable has class imbalance. In this case, few people had diabetes, meaning a model could still be relatively accurate by only predicting that people don't have diabetes. Therefore, between accuracy and AUC, it would be better to compare AUC. I plotted both accuracy and AUC to visually inspect which model had the highest performance. I also found the model with the highest AUC and accuracy as this would be the model with the best performance (i.e. the best model at predicting diabetes).
c. As mentioned above, I found a large class imbalance for the outcome variable. The figure on the left shows that there were far fewer people diagnosed with diabetes than not diagnosed. Additionally, as shown on the right, the adaBoost model had both the highest AUC and accuracy across the models. The AUC, however, was relatively the same at around 0.8 for all models except for the single decision tree which had an AUC of about 0.6. The accuracy of adaBoost is also virtually the same as that of the random forest and SVM models.




d. While all models had roughly the same AUC (except for the single decision tree which had a much lower performance), adaBoost had a slightly higher AUC as well as a higher accuracy than the other models. Additionally, because of the class imbalance, it makes more sense to compare AUC than accuracy. In this case,

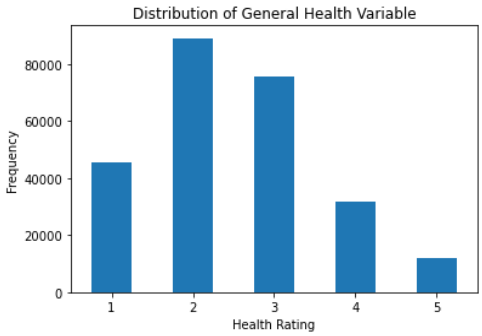
adaBoost is the best model at predicting diabetes because it has the highest AUC - even though the difference between it and the other models is minimal.

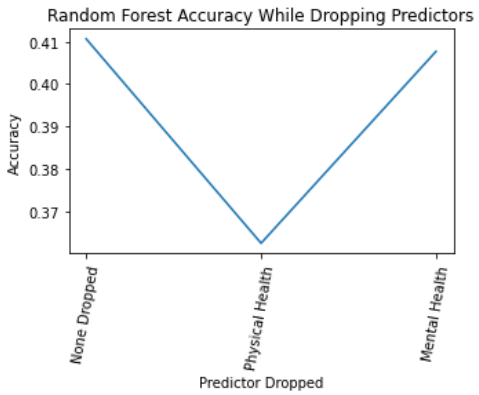
**B. Tell us something interesting about this dataset that is not already covered by the questions above and that is not obvious.**

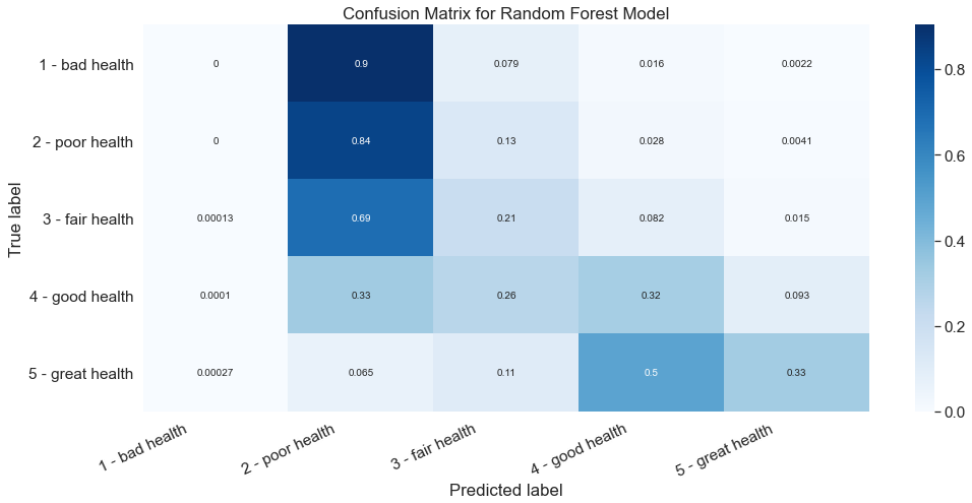
a. To further analyze the data, I wanted to see if the General Health variable could be predicted using Physical Health and Mental Health and which feature was stronger. To do this, I built a random forest classifier to predict General Health using Physical and Mental Health as predictors. I split the data into a training and test set with the test set being 30% of the total data. The random forest model used 500 estimators, bootstrapping, and the Gini Index criterion. I then found the accuracy of the model and plotted a confusion matrix to show the model's performance. Lastly, I dropped each of the two features one by one, computed the new models' accuracies, and plotted them across models to see which feature was more important.

b. I used a random forest model because, since the outcome variable wasn't binary, it makes more sense to use multiclass classification and random forest classifiers are generally robust and straightforward to implement. I used 500 estimators as a larger number of estimators helps the model converge (at the expense of runtime). I also used accuracy to measure model performance as it allowed me to evaluate performance across all five outcome classes, whereas AUC would need to be computed for one vs. all classes or one vs. one. I plotted the confusion matrix to evaluate performance across classes and see which classes were more likely to be correctly or incorrectly classified in the model. Similarly to the other questions, I found feature importance by seeing which predictor dropped model performance the most when removed.

c. As shown below, there is an imbalance between classes for General Health responses. Specifically, most people responded with 2 or 3, and very few responded with 5. Additionally, the confusion matrix of the model shows a poor performance, with only the response '5' having a somewhat consistent prediction. This is reflected in the overall accuracy of 0.41062. Physical Health also sharply dropped the model's accuracy when removed, while removing Mental Health only slightly lowered the model's performance.




d. Without accounting for class imbalance, a random forest model predicting General Health from Mental and Physical Health performed poorly (accuracy = 0.41). The confusion matrix shows that the model was more likely to predict a datapoint as having a health rating of 2 than other values - which was the most frequent value in the dataset. Interestingly, dropping physical health dropped model performance much more than mental health. This suggests that in this dataset physical health corresponds more with an individual's rating of their overall health than mental health.

## Relevant Code Output (for your reference)

```
Index(['Diabetes', 'HighBP', 'HighChol', 'BMI', 'Smoker', 'Stroke',
       'Myocardial', 'PhysActivity', 'Fruit', 'Vegetables', 'HeavyDrinker',
       'HasHealthcare', 'NotAbleToAffordDoctor', 'GeneralHealth',
       'MentalHealth', 'PhysicalHealth', 'HardToClimbStairs', 'BiologicalSex',
       'AgeBracket', 'EducationBracket', 'IncomeBracket', 'Zodiac'],
      dtype='object')
(253680, 22)
                      Diabetes    HighBP  ...  IncomeBracket    Zodiac
Diabetes              1.000000  0.263129  ...      -0.163919 -0.000197
HighBP                0.263129  1.000000  ...      -0.171235 -0.002629
HighChol              0.200276  0.298199  ...      -0.085459  0.001052
BMI                   0.216843  0.213748  ...      -0.100069 -0.001932
Smoker                0.060789  0.096991  ...      -0.123937  0.000975
Stroke                0.105816  0.129575  ...      -0.128599 -0.001933
Myocardial            0.177282  0.209361  ...      -0.141011 -0.001232
PhysActivity         -0.118133 -0.125267  ...       0.198539 -0.000842
Fruit                -0.040779 -0.040555  ...       0.079929 -0.002255
Vegetables           -0.056584 -0.061266  ...       0.151087 -0.000732
HeavyDrinker         -0.057056 -0.003972  ...       0.053619 -0.001216
HasHealthcare         0.016255  0.038425  ...       0.157999  0.000502
NotAbleToAffordDoctor  0.031433  0.017358  ...      -0.203182  0.001856
GeneralHealth         0.293569  0.300530  ...      -0.370014  0.001435
MentalHealth          0.069315  0.056456  ...      -0.209806  0.002008
PhysicalHealth        0.171337  0.161212  ...      -0.266799  0.001977
HardToClimbStairs     0.218344  0.223618  ...      -0.320124  0.003291
BiologicalSex         0.031430  0.052207  ...       0.127141  0.001671
AgeBracket            0.177442  0.344452  ...      -0.127775 -0.002632
EducationBracket     -0.124456 -0.141358  ...       0.449106 -0.001313
IncomeBracket        -0.163919 -0.171235  ...       1.000000 -0.003201
Zodiac               -0.000197 -0.002629  ...      -0.003201  1.000000

[22 rows x 22 columns]
0    218334
1     35346
Name: Diabetes, dtype: int64

PREDICTING WITH LOGISTIC REGRESSION:

Best Predictor: BMI
AUC of full model: 0.8210348275267177

PREDICTING WITH SUPPORT VECTOR MACHINES:
Optimal Slack Value: 0.125

Best Predictor: BMI
AUC of full model: 0.8207875021508929
```

PREDICTING WITH SINGLE DECISION TREE:

Best Predictor: BMI
AUC of full model: 0.601114227021972

PREDICTING WITH RANDOM FOREST:
Highest AUC = 0.810437660884795
Optimal Parameters: {'max_features': 0.25, 'max_samples': 0.5, 'n_estimators': 100}

Best Predictor: BMI
AUC of full model: 0.8105358049186033

PREDICTING WITH ADABOOST:
Highest AUC = 0.825708292144148
Optimal Parameters: {'learning_rate': 0.1, 'n_estimators': 500}

Best Predictor: BMI
AUC of full model: 0.8248537025539403

BEST MODEL:

Highest Accuracy: AdaBoost
Highest AUC: AdaBoost

OTHER ANALYSES:

Correlation Between General Health and Physical/Mental Health:
PhysicalHealth 0.524364
MentalHealth 0.301674
Name: GeneralHealth, dtype: float64

Accuracy: 0.41062230631767055
Best Predictor of General Health: Physical Health