FML Homework 3 - Erin Choi

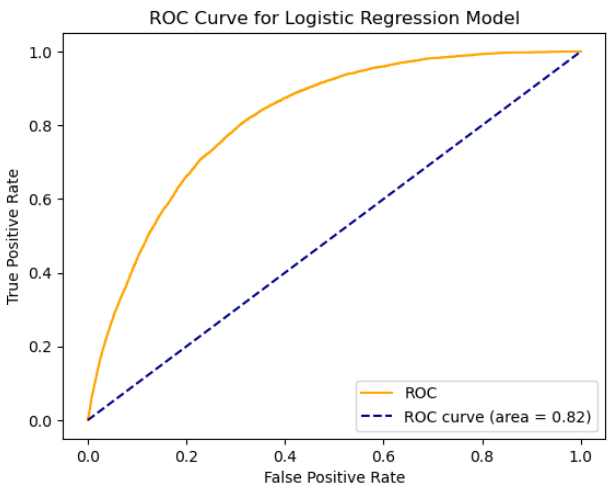
Data Cleaning and Preprocessing

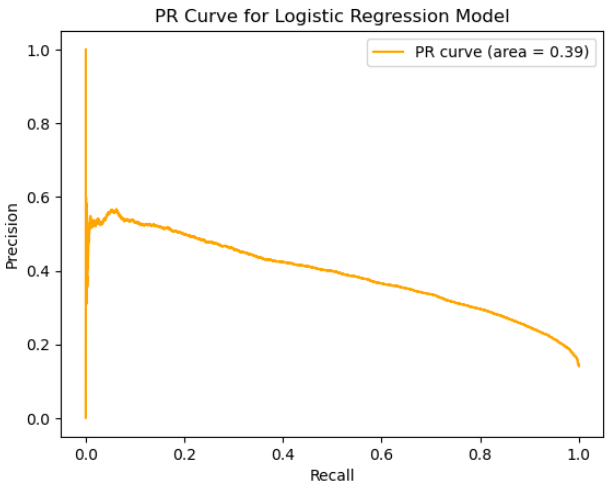
After loading the data, I created lists for each type of variable - binary, ordinal, categorical, and continuous numeric - to aid in preprocessing. I dropped the Zodiac variable as I thought it would have a negligible effect on diabetes classification. I changed the gender (BiologicalSex) variable to be encoded as 0 and 1 rather than 1 and 2 to make it consistent with the rest of the binary variables. I then scaled the ordinal and continuous features using MinMaxScaler to get their ranges similar to that of the binary variables. I checked the class balance before separating the feature and target variables and splitting them into training and test sets, with 20% of the data set aside for testing. There was a large class imbalance, with most of the data belonging to the class with no diabetes, so I used StratifiedShuffleSplit to make sure the class proportions were the same in the training and test sets.

Question 1

I built a logistic regression model with the class_weight parameter set to balanced because of the class imbalance. The model failed to converge with the default solver, so I changed the solver to liblinear. I found the AUROC, AUPRC (computed using average precision score), accuracy, F1 score, and confusion matrix for the model. Using this model, I looped through all the predictors and ran the model, dropping the current predictor from the data for each iteration. For each model with one dropped predictor, I saved the AUROC and AURPC. The best predictor was determined to be the predictor that dropped each of the two metrics the most when it was excluded from the data. AUC was used as the metric for choosing the best predictor because is it a better indicator of model performance using imbalanced data than accuracy.

The logistic regression model resulted in an AUROC of **0.818**, AUPRC of 0.391, accuracy of 0.731, and F1 score of 0.440. The plots for the ROC curve and PR curve are below. The best predictors were **GeneralHealth,** which dropped the model's AUROC by 0.015, and **BMI**, which dropped AURPC by 0.027.





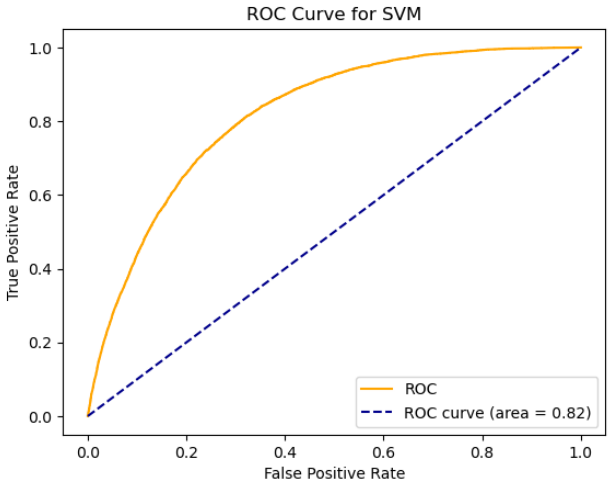
```
Confusion matrix:
 [[31707 11960]
 [ 1700  5369]]
```

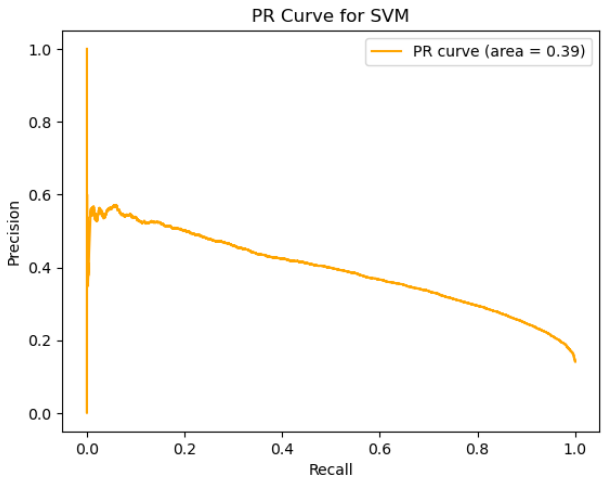
The overall AUC of 0.818 is fair and indicates that the model does a decent job at separating the 0 and 1 classes. The accuracy of 0.731 is a bit low but somewhat acceptable. However, the precision and recall are not the best, as indicated by the lower AUPRC and F1 score. This comes from the high false positive rate of 0.274; the resulting confusion matrix shows that the model classifies many units without diabetes as having diabetes. While this is a problem, it is better than having more false negatives in this case, as it would be worse to not catch people who actually have diabetes than to misclassify people who don't actually have diabetes. Additionally, it makes sense that BMI is a good predictor of diabetes, as it is known that obesity makes one more likely to develop diabetes. While it also makes sense that general health is a good predictor, I would be careful depending too much on this feature in this particular dataset, as its values come from self-assessments.

Question 2

I built an SVM, again with the class_weight parameter set to balanced to handle the class imbalance. I set the dual parameter to false to use the squared hinge loss function rather than hinge loss, as the model performed better with this setting. I was not able to improve the model by changing the C parameter for the error term. Again, I found the AUROC, AUPRC, accuracy, F1 score, and confusion matrix for this model. I used the SVM to determine the best predictors using the same drop-one-predictor procedure described in Question 1.

The SVM model resulted in an AUROC of **0.818**, AUPRC of 0.392, accuracy of 0.727, and F1 score of 0.439. The plots for the ROC curve and PR curve are below. The best predictors were **GeneralHealth**, which dropped the model's AUROC by 0.016, and **BMI**, which dropped AURPC by 0.028.






```
Confusion matrix:
 [[31491 12176]
 [ 1657  5412]]
```

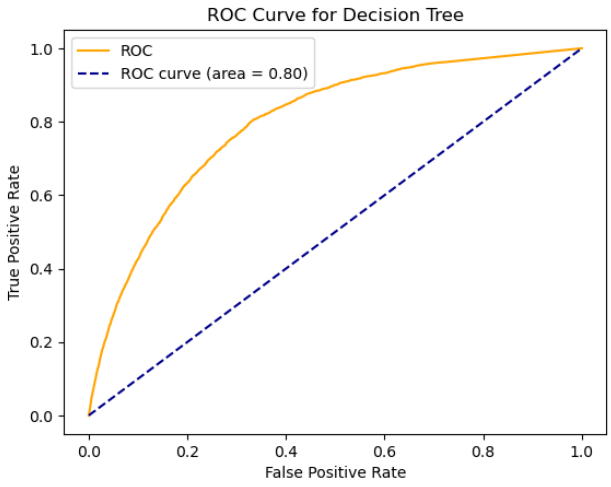
This model had almost identical performance to the logistic regression model, with differences of 0.001 for the AUPRC and F1 score. The overall AUC and accuracy are, again, indicative of decent performance in separating classes. Leaving out the best predictors dropped the SVM's performance by 0.001 more than it decreased logistic regression performance for both AUROC and AUPRC, which means the SVM

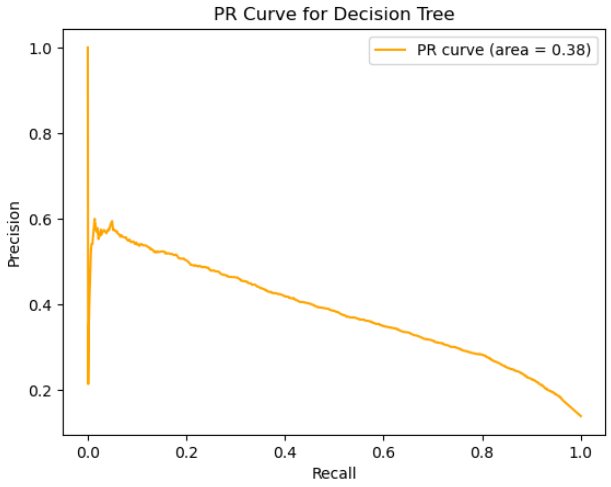
relies on GeneralHealth and AUROC very slightly more than logistic regression. Again, a high false positive rate (0.279) results in a low AUPRC and F1 score, but this is better for predicting diabetes than a higher false negative rate for reasons previously explained in Question 1; the false positive rate is greater than the false negative rate of 0.234. The interpretation of the best predictors stays consistent with the explanation in Question 1 as well.

Question 3

I built a decision tree, initially setting the class_weight parameter to balanced to handle the class imbalance. I attempted to improve the tree's performance through hyperparameter tuning; changing the default criterion (Gini) to entropy increased the AUC, then I refit the model several times with different min_samples_split values to determine the number that yielded the highest AUC. I ended up using 201 because it resulted in the highest accuracy within a range of values that gave me the highest AUC. I found the AUROC, AUPRC, accuracy, F1 score, and confusion matrix for my final model then used it within my drop-one-predictor loop to determine the best predictors, following the procedure described in Question 1.

The tree that I began with, which used class_weight='balanced' as its only parameter, yielded an AUC of 0.60, AURPC of 0.19, accuracy of 0.797, and F1 score of 0.30. By changing the split criterion and the minimum number of samples required for a node split, I brought the AUC performance up. The final tree resulted in an AUROC of **0.799**, AUPRC of 0.383, accuracy of 0.713, and F1 score of 0.424. The plots for the ROC curve and PR curve are below. Again, the best predictors were **GeneralHealth**, which dropped the model's AUROC by 0.018, and **BMI**, which dropped AURPC by 0.028.





The single decision tree performed similarly to but worse than the previous two models. This was expected, as single trees are weak learners, which generally have slightly better performance than random guessing. Hyperparameter tuning did improve the tree's performance significantly, bringing both AUROC and AUPRC up by almost 0.2. The accuracy decreased by about 0.084 between the initial tree and the final tree; I prioritized optimizing the AUC because it is a better measure of performance on imbalanced data, and I did sacrifice some accuracy in the process. The confusion matrix

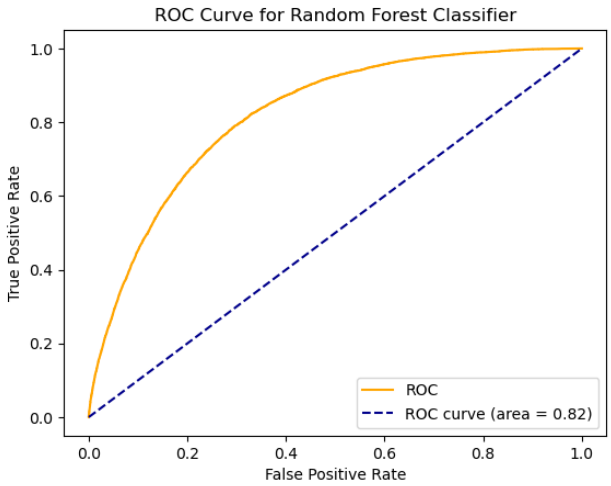
```
Confusion matrix:
 [[30790 12877]
 [ 1710  5359]]
```

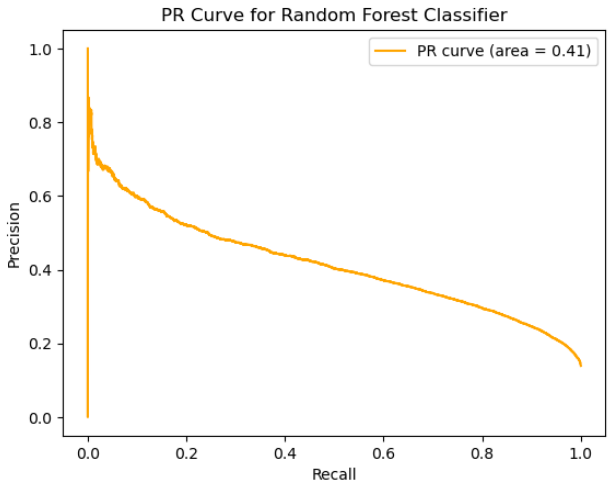
shows that the tree did a worse job classifying both positive and negative units correctly, as the numbers for true positives and true negatives have decreased from those for the previous two models. The tree is still yielding a higher false positive rate (0.295) than false negative rate (0.242), which is better for our diabetes classification case as per the explanation in Question 1, and the interpretation of the best predictors remains the same again.

Question 4

I began building a random forest model by setting only the class_weight parameter to balanced to handle the class imbalance. I tried to improve the model's performance by changing several hyperparameter values. Changing the criterion from its default of Gini did not affect the results much, and decreasing the proportion of max_features to less than the whole sample actually worsened performance. Increasing the number of estimators did not change the performance much either but increased computing time greatly. I was able to improve performance by increasing the min_samples_split value from 2 to 30; this value kept the highest accuracy score among the values that yielded the highest AUC. I found the AUROC, AUPRC, accuracy, F1 score, and confusion matrix for my final model and followed the drop-one-predictor procedure from Question 1 to determine the best predictors using random forest.

My initial random forest model with only class_weight set to balanced resulted in an AUROC of 0.79, AUPRC of 0.36, accuracy of 0.857, and F1 score of 0.24. Tuning the min_samples_split hyperparameter improved AUC performance, yielding an AUROC of **0.820**, AUPRC of 0.414, accuracy of 0.782, and F1 score of 0.458. The plots for the ROC curve and PR curve are below. Once again, the best predictors were **GeneralHealth**, which dropped the model's AUROC by 0.018, and **BMI**, which dropped AURPC by 0.034.






```
Confusion matrix:
 [[35002  8665]
 [ 2391  4678]]
```

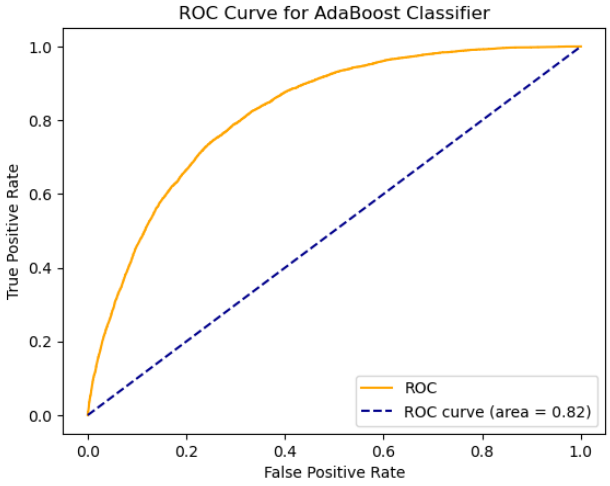
The random forest had the highest values for all four calculated metrics compared to all previous models, including accuracy, which dropped a lot as a result of prioritizing high AUC values during hyperparameter tuning. The AUROC and accuracy are both fair and indicative of decent class separation by the model, while the AUPRC and F1 score remain quite low. The confusion matrix looks noticeably different from that of previous models, with more true negatives, true positives, and false negatives, and

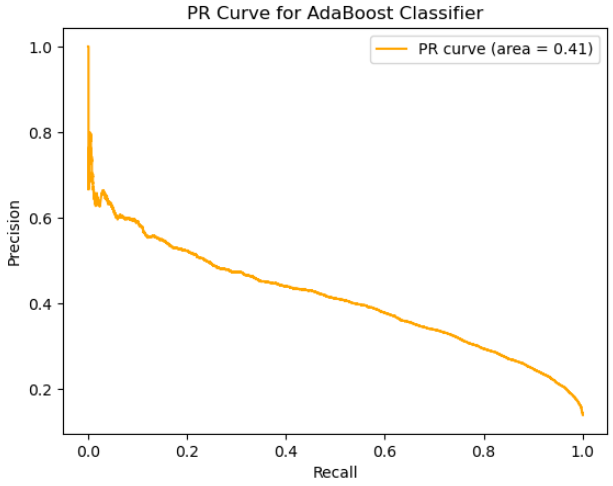
upon further inspection, this model actually has a much higher false negative rate (0.338) than false positive rate (0.198) unlike the other models. This is not good for classifying those who do and do not have diabetes, as this model would classify those who have diabetes as not having diabetes at a higher rate than classifying people who don't have diabetes as having it. The higher false negative rate has very dangerous implications for this use case, so even with the best performance in terms of AUC, the random forest model may not really be the "best" model. The interpretation of the two best predictors remains the same as in Question 1.

Question 5

I began building the AdaBoost classifier as I did all other models, with the only parameter being class_weight set to balanced. I set the decision tree's max depth to 1, as this is the default estimator setting for AdaBoost. To optimize AUC values, I increased the tree's min_samples_split to 100 and changed the AdaBoost model's number of estimators to 100. I also attempted to vary the learning rate, but it increased computing time by a lot without a lot of improvement. With these settings, I found the model's AUROC, AUPRC, accuracy, F1 score, and confusion matrix and applied the drop-one-predictor procedure from Question 1 to determine the best predictors.

My initial AdaBoost classifier with only class_weight set to balanced resulted in an AUROC of 0.77, AUPRC of 0.29, accuracy of 0.810, and F1 score of 0.34. Hyperparameter tuning to improve AUC yielded an AUROC of **0.822**, AUPRC of 0.412, accuracy of 0.731, and F1 score of 0.442. The plots for the ROC curve and PR curve are below. This time, the best predictor based on both metrics was **BMI**, which dropped the model's AUROC by 0.016 and AURPC by 0.034.





The AdaBoost model had a decent AUROC and fair accuracy, again indicating reasonable class separation capabilities. Accuracy again dropped as I tried to optimize AUROC. AUPRC and F1 score were low for this model as well, but the false positive rate of 0.274 is greater than the false negative rate of 0.236, which is favorable when predicting diabetes classes. BMI being the best predictor based on AUROC and AUPRC makes sense because of the positive relationship between obesity and diabetes risk.

```
Confusion matrix:
 [[31689 11978]
 [ 1665  5404]]
```

Extra Credit Part 1

Combining my results for all the above questions, I could easily compare the performance metric values of all the models. AdaBoost performed best in terms of AUROC, while the random forest model produced the highest values for AUPRC, accuracy, and F1 score.

	Model	AUROC	AUPRC	Accuracy	F1
0	LR	0.818136	0.391079	0.730763	0.440118
1	SVM	0.817908	0.392053	0.727353	0.438983
2	DTC	0.798959	0.382944	0.712492	0.423553
3	RF	0.819553	0.413838	0.782088	0.458358
4	ABC	0.821748	0.411558	0.731098	0.442027

However, as mentioned in Questions 4 and 5, the random forest model yielded a higher false negative rate than false positive rate, while the AdaBoost classifier's false positive rate was greater than its false negative rate. As detailed in Question 1, having a higher false positive rate is less consequential than a higher false negative rate when predicting classes for diabetes. It is very dangerous to tell people they don't have diabetes when they actually do at a higher rate than the other way around.

Additionally, as mentioned in Question 1, I prioritized maximizing AUROC for each model when hyperparameter tuning (if tuning made changes at all), since it is a better measure of performance for models based on data with imbalanced classes than accuracy. Ultimately, the best model for predicting diabetes using this dataset is the **Adaboost classifier,** which has both a higher AUROC and a higher false positive rate than false negative rate.