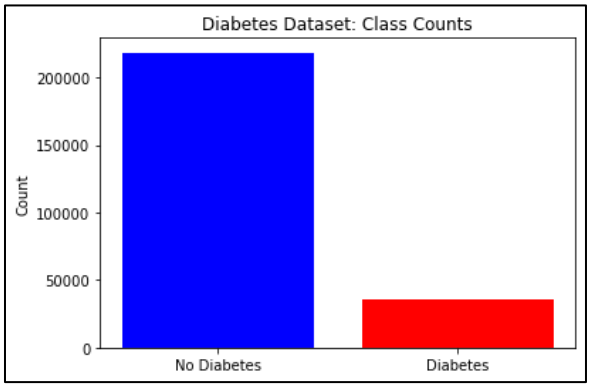
Tomisin Adeyemi - FML HW 3

Preamble: I used a test size of 0.2 for all my models, as well a random state of 1234. The graph below shows the class counts for Diabetes in the dataset. Because of the high class imbalance, I made sure to specify the class_weight parameter as balanced for each model I used. In addition, to find the best predictor for each model, I created multiple models, dropping a single predictor each time, then looked out for which predictor dropped the AUROC/AUPR the most. I looked at the AUROC mainly because each question asked for the AUC of the best performing model. The AUPR was mainly a sanity check because Precision-Recall is a better metric for highly imbalanced datasets. In this report, X refers to a set of predictors, and y refers to the outcome variable (Presence of Diabetes).



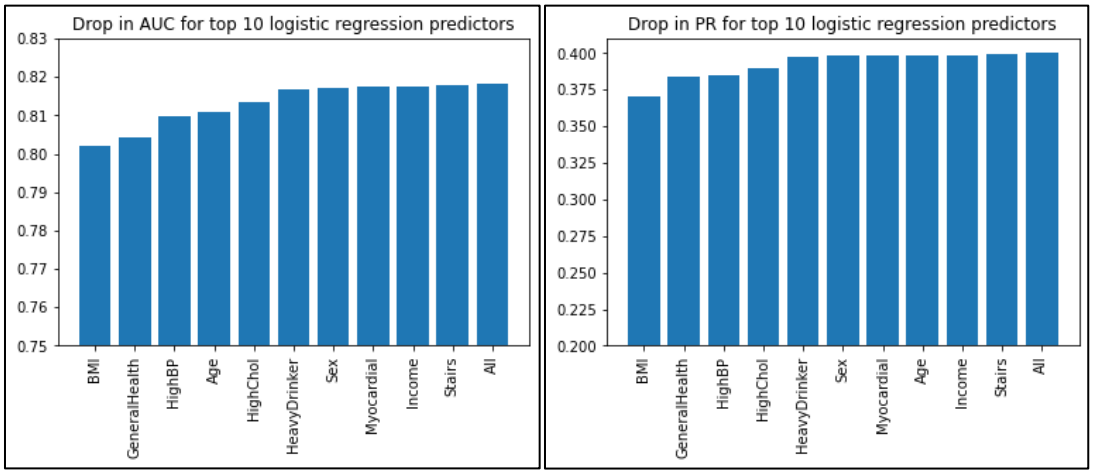
1. Build a logistic regression model. Doing so: What is the best predictor of diabetes and what is the AUC of this model?

I split the data into training and test sets, then created a LogisticRegression model with a liblinear solver and balanced class weights. I then computed the predicted probabilities for the test values of X, and used the predicted probabilities of the positive class to pick a threshold by picking the threshold that maximized the f1 score. (This resulted in a threshold of 0.6, as opposed to Scikit-Learn's default of 0.5). Then, for each predictor in X, I created a logistic regression model without the feature and stored the AUROC/AUPR values in a dataframe. I then plotted the top 10 predictors with the highest drop in AUROC/AUPR. The best predictor was the predictor whose removal resulted in the largest drop for these metrics. I computed the AUROC and AUPR using scikit-learn's metrics.roc_curve() and metrics.average_precision_score (approximates AUPR) with the test y variable and the predicted probabilities for the positive class as parameters.

I decided to pick AUROC/AUPR as the main metrics to assess model performance. Because this is a highly imbalanced dataset, and because it is very beneficial to detect diabetes itself rather than the percentage of correctly classified cases as whole, using just accuracy as a performance metric is not very helpful. This applies to questions 2 to 5 as well. In addition, dropping predictors as opposed to testing each predictor individually lets the model take advantage of correlated features, and avoids the effects of trying to use lone predictors that have very low prediction power (negligible metric scores). This also applies to questions 2 to 5.

The initial AUROC for the entire model (no predictors dropped) was 0.818, and the AUPR was 0.400. The variable that resulted in the biggest drop in AUROC was BMI, resulting in a drop to 0.802. For AUPR, the

predictor that resulted in the largest drop was also BMI, resulting in a drop to 0.371. Below is the graph for the top 10 logistic regression predictors in terms of AUROC/AUPR.

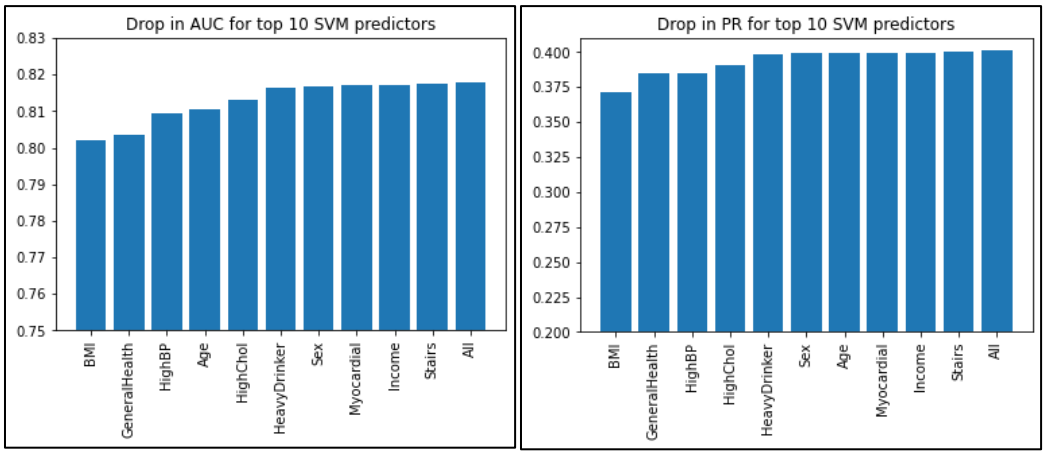


Even though BMI had the lowest drop in AUROC, the drop is relatively small and is still a very good AUROC score to have, i.e. the model is still relatively good at predicting Diabetes when evaluated using AUROC. The AUPR score though is not very good, meaning that the model has an uncomfortable number of false positives.

2. Build a SVM. Doing so: What is the best predictor of diabetes and what is the AUC of this model?

The approach used is like the one used in logistic regression above. I used the svm.LinearSVC model in scikit-learn, setting dual to False (to make sure the algorithm converges, and to use the squared hinge loss function) and the class weights to be balanced. I decided to use LinearSVC, as the other SVM functions are not recommended (by Scikit-learn documentation) for a dataset as big as this one. I used AUROC/AUPR to evaluate model performance as stated in the preamble and number 1.

For the SVM model using all the predictors, the AUROC was 0.818, and the AUPR was 0.401. The predictor that resulted in the highest drop in AUROC was again BMI, resulting in a drop to 0.802. For the AUPR, it was again BMI, resulting in a drop to 0.371. (These results are uncomfortably similar to the results for logistic regression in part 1)

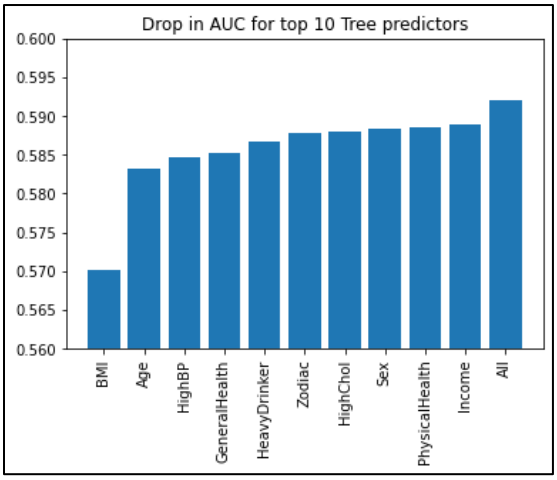


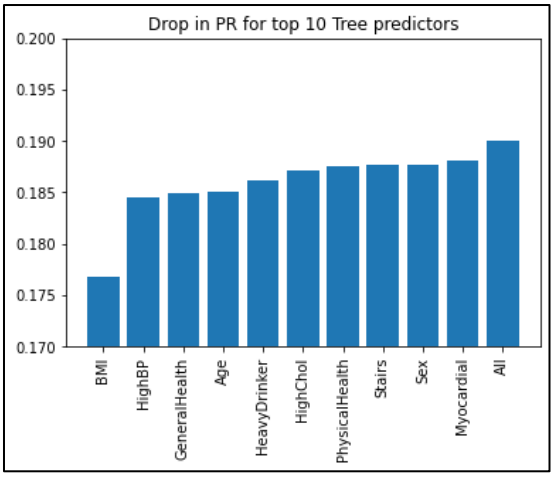
Perhaps if experimentation with other kernels (radial, polynomial, etc) was possible, we could obtain better metrics in terms of AUPR for SVM. But for now, the conclusion is the same as logistic regression above.

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

I used the tree.DecisionTreeClassifier scikit-learn function, making sure the class weight is balanced, and setting the criterion to entropy. I decided to use entropy to quantify leaf impurity, rather than the Gini index, because it resulted in higher AUROC/AUPR scores. Again, AUROC/AUPR was used as the performance metric for reasons previously stated.

The AUROC for the entire model (before any predictors are dropped) was 0.593, and the AUPR 0.191. Again, the predictor that resulted in the highest drop for AUROC and AUPR was BMI, with scores of 0.570 and 0.178 respectively.



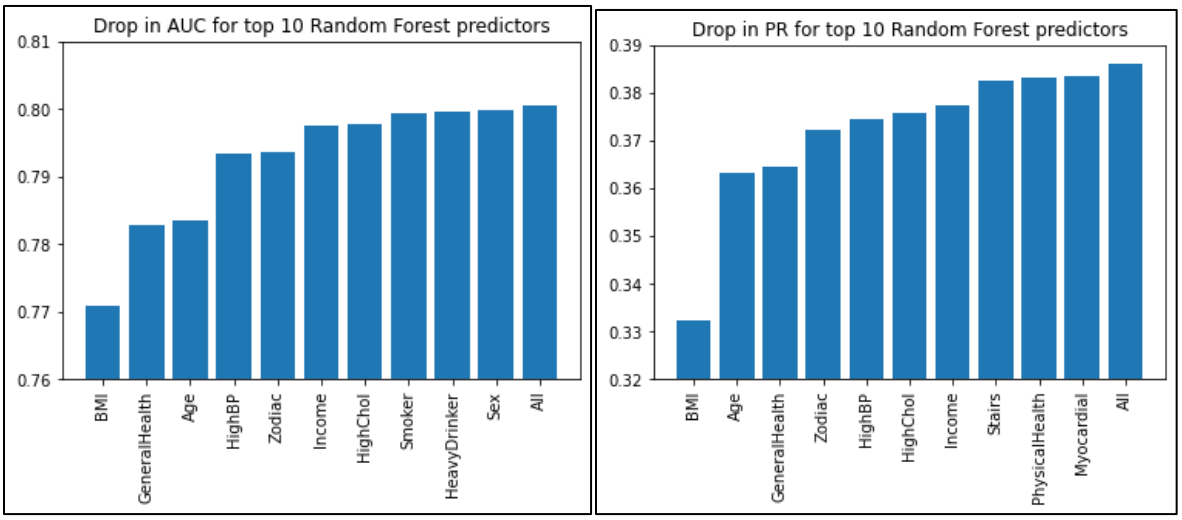


It is not surprising that AUPR/AUROC scores are very low for the decision tree (turn out to be the lowest scores across all models): an AUROC score of 0.593, suggests that the model is very close to having the same output as a model that just outputs random values. Single decision trees are known to be weak learners, they tend to overfit to the training data, not generalizing properly. Hence, these results are expected.

4. Build a random forest model. Doing so: What is the best predictor of diabetes and what is the AUC of this model?

I used the RandomForestClassifier in scikit-learn, using the following hyperparemeters: balanced class weight, n_estimators=100, max_samples=0.5, max_features=0.5, bootsrap=True and the criterion as gini. I initially tried to do hyperparameter tuning using GridSearchCV, but that was taking way too long so I just played around with the parameters and picked the ones that performed best. I used AUROC/AUPR as performance metrics for reasons explained previously.

The AUROC for the model with all predictors is 0.802, and the AUPR is 0.386. Again, BMI is the predictor that results in the highest drop for both AUROC and AUPR, resulting in scores of 0.771 and 0.332 respectively.

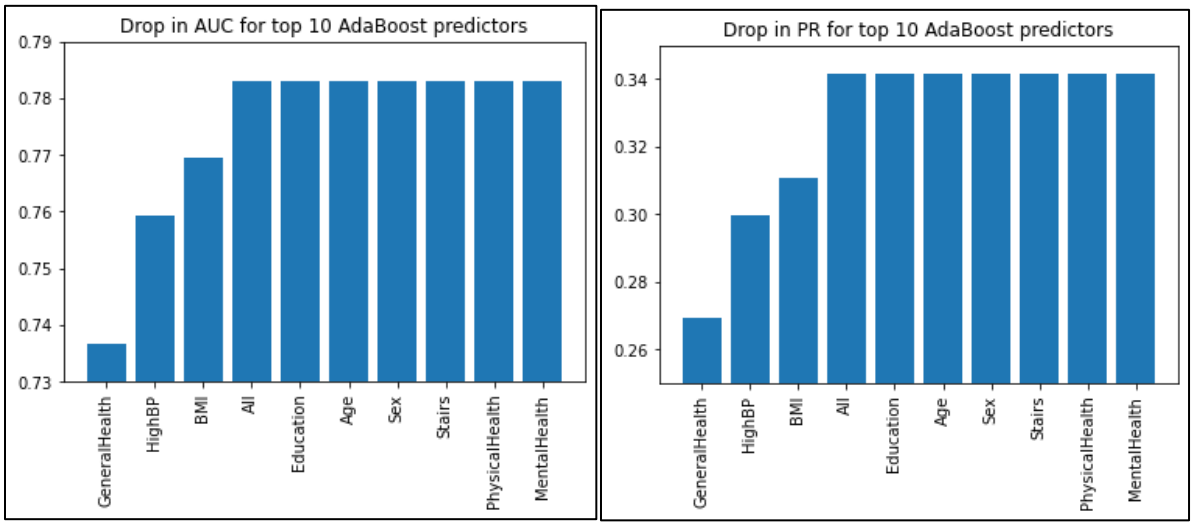


It is not surprising that this model performs better than the single tree, as the random forest uses the ensemble of weak learner trees to form a strong learner. It still does not perform as well as SVM and logistic regression.

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

I used the AdaBoostClassifier from Scikit-learn, using a tree.DecisionTreeClassifier with a max depth of one (because Adaboost uses stumps), and balanced class weights. I used 300 estimators and a learning rate of 1.5. I then found the AUROC/AUPR performance metrics for reasons explained previously.

The AUROC and AUPR of the model using all predictors is 0.783 and 0.341 respectively, slightly worse than the Random Forest classifier. This time, the predictor that resulted in the largest increase in AUC and AUPR was GeneralHealth, resulting in values of 0.737 and 0.269 respectively. Another interesting thing is that the AUROC and AUPR stayed the same for all predictors apart from GeneralHealth, HighBP and BMI. The graph below shows the results.



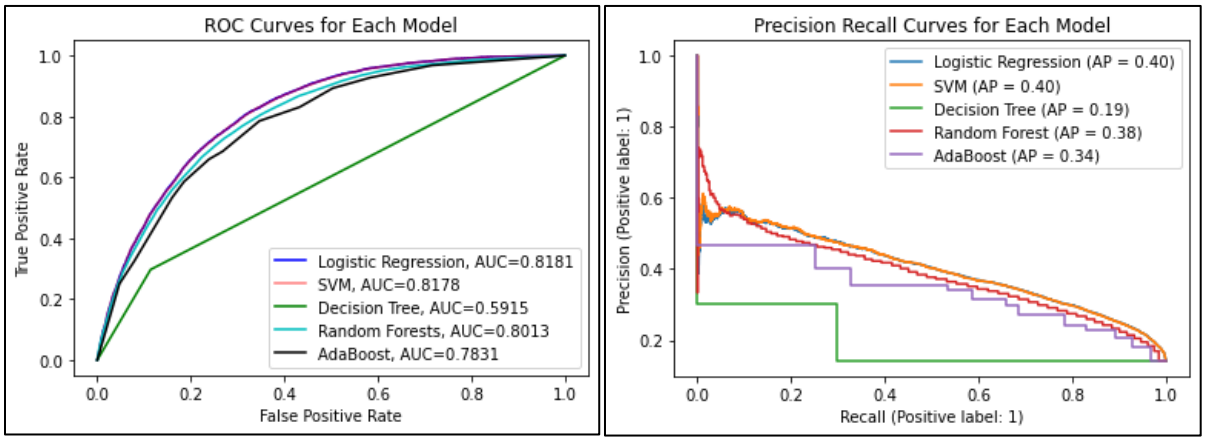
The previous points on interpretation of AUROC/AUPR still stand. Perhaps the reason that the AUROC/AUPR scores stay the same for most predictors is that AdaBoost is based on weighted stumps, which focus more on features that are important and makes decisions based on that. It is also very robust to noise so perhaps it considers the columns other than GeneralHealth, HighBP and BMI as noise. This suggests that according to the AdaBoost Algorithm, GeneralHealth, HighBP and BMI are the most important variables for predicting diabetes.

Extra credit: a) Which of these 5 models is the best to predict diabetes in this dataset?

I decided to recompute the AUROC and AUPR for each model (using the same hyperparameters used before) and plot the AUROC and AUPR for each model. For the AUROC I used the metrics.roc_curve metric with the y_test variable and predicted probability of the positive class as input. I then plotted the false positive rate against the true positive rate for each model. For the AUPR, I simply called the metrics.plot_precision_recall_curve() with the respective models, and the X and y test variables as parameters.

As before, I included the AUROC curves because questions 1 to 5 ask for it. I used the AUPR for more of a sanity check as the classes are highly imbalanced.

The plots below show the AUROC and AUPR for each model.

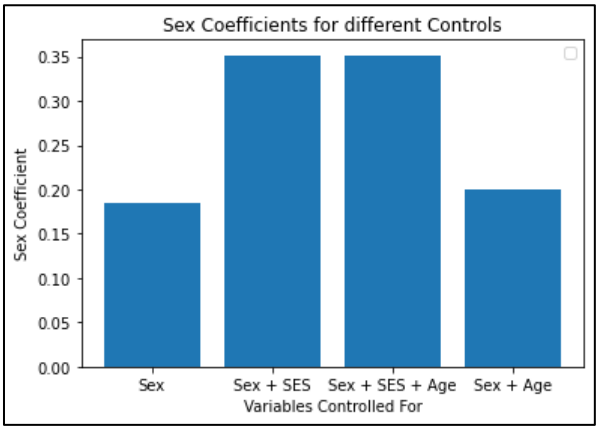


From these diagrams, we can see that Logistic Regression and SVM are the best models in terms of AUROC and AUPR: the curves for both models (in bothe the AUROC and AUPR) overlap!

b) Tell us something interesting about this dataset that is not already covered by the questions above and that is not obvious

Questions 1 to 5 showed that BMI and health-related predictors were useful to the model to produce good AUROC and AUPR scores. I decided to investigate another predictor, Biological Sex, and see if there were any appreciable betas with and without controlling for other factors. I controlled for Socio-Economic Status (Education Bracket and Income Bracket), as well as Age. I did this mainly by including/not including this variables as predictors. I encoded the Male and Female variables as 0 and 1, meaning that the Female sex served as the positive class. After, I simply inspected the betas by using the model.coef_ method and checking the appropriate entry for Sex.

The graph below shows the results. Sex as the lone predictor had a beta of 0.184, Sex had a beta of 0.352 controlling for SES, a beta of 0.351 controlling for SES and Age, and a beta of 0.2 just controlling for age.



The betas values can still be interpreted as follows. The sex coefficient represents the log of the odds ratio that associates sex with the risk of diabetes. Let's take the highest beta: 0.352. $e^{\beta} = e^{0.352} = 1.422$, meaning that 1.422 is the odds ratio that associates sex with the risk of diabetes after controlling for SES. Since Female sex is the positive class, this means that people of Female sex have 1.42 times the odds of the Male Sex of having diabetes, after controlling for Socio-Economic Status. Alternatively, there is a 42% greater risk of getting diabetes if one is of Female Sex, controlling for SES.

This is a pretty big deal, and should elicit efforts to provide better access to diabetes treatments for people of the Female sex.

.