Predicting Diabetes using Machine Learning

"All models are wrong, but some are useful."

-George E.P. Box

This quote captures the essence of this project. From not being able to import the dataset directly from Kaggle to spending what felt like endless hours tuning hyperparameters with gridsearchCV, this project has been a great learning experience. To make a long story short, a complex model does not always mean a better model. The challenge is coming up with the right model that fits in this eight-dimensional space (8 corresponds with the number of variables in the study that are not response variables) with accuracy and precision. It's currently not possible to create an 8-dimensional scatterplot and graph a regression hyperplane, so modeling is used to "guess" where the hyperplane is drawn between diabetics and non-diabetics. It is traditional for data scientists and data enthusiasts to compete on getting more accurate models for a dataset, but therein lies the first problem of this study. Accuracy should not be the main goal when disease is concerned. Rather than accuracy, there should be more focus on precision. The goal of high precision is to diagnose patients with diabetes only when it is truly confirmed they have diabetes. It can be dangerous to treat a disease that is not present (False positive). Usually, it is better to test a patient that is negative (False negative) than to treat a patient that is actually negative (False positive). Despite this fundamental flaw present throughout the study, keep reading and learn from the mistakes made here.

The Hypothesis

"Will using Machine Learning techniques increase accuracy of the model by at least 5% (model accuracy of at least 87.98%), thus significantly improving upon the logistic regression model?"

Notice that accuracy is mentioned here rather than precision. Also, 5% is a very steep increase. Given that the models are about equally difficult to implement, a 1-2% increase would suffice. Also, it is mentioned in the proposal that the import of data would be done using BeautifulSoup. This was attempted, but failed due to a missing comma or two in the csv file. Once the csv was fixed and properly imported, the next step was to clean the data. This was arguably the most important step in the entire project, and the first attempt led to abysmal scores.

The Cleaning

A removed outlier here, a removed outlier there, a few mean imputations there, and the cleaning was set. In reality, the most time should have been spent on this step to determine what is truly an outlier or an odd result. Had there been more time for this project, consulting with medical professionals on what would be an unusual result for each variable based on diabetes data would have been helpful. The second cleaning attempt copies the original author's cleaning steps. All the original author did was remove rows where the patient had a glucose of 0 and was diabetic. Most of the time was spent in the modeling step of this project, which wouldn't have happened if the data would have been cleaned properly on the first attempt. However, it does take a while to tune hyperparameters.

The Modeling

The first classifiers were created using no hyperparameter tuning (not shown on the notebook). The scores were around 60% accuracy, which is quite low compared to the 82% accuracy the logistic regression model got. Two days were spent hyperparameter tuning using gridsearchCV, and it resulted in about a 3-4% increase in accuracy. How could the accuracies be so low compared to logistic regression? Looking back at the original author's work, the cleaning steps were much simpler. Again, given more time, consulting a medical professional to get the cleaning just right would have been ideal, but there was quite a bit of fixing to do with the project. Starting almost completely over several days in is risky, but rewarding. After five days of hyperparameter tuning, the random forest model's accuracy increased to just below the logistic regression accuracy of 80.5%. The exact SVM and logistic regression models couldn't be reproduced exactly due to a lack of random_state. The random_state parameter seeds the classifier so that the model can be reproduced outside the author's computer and will not change values every time the notebook is run. With a lower accuracy of 80.5% for the logistic regression, none of the three models created could surpass the logistic regression.

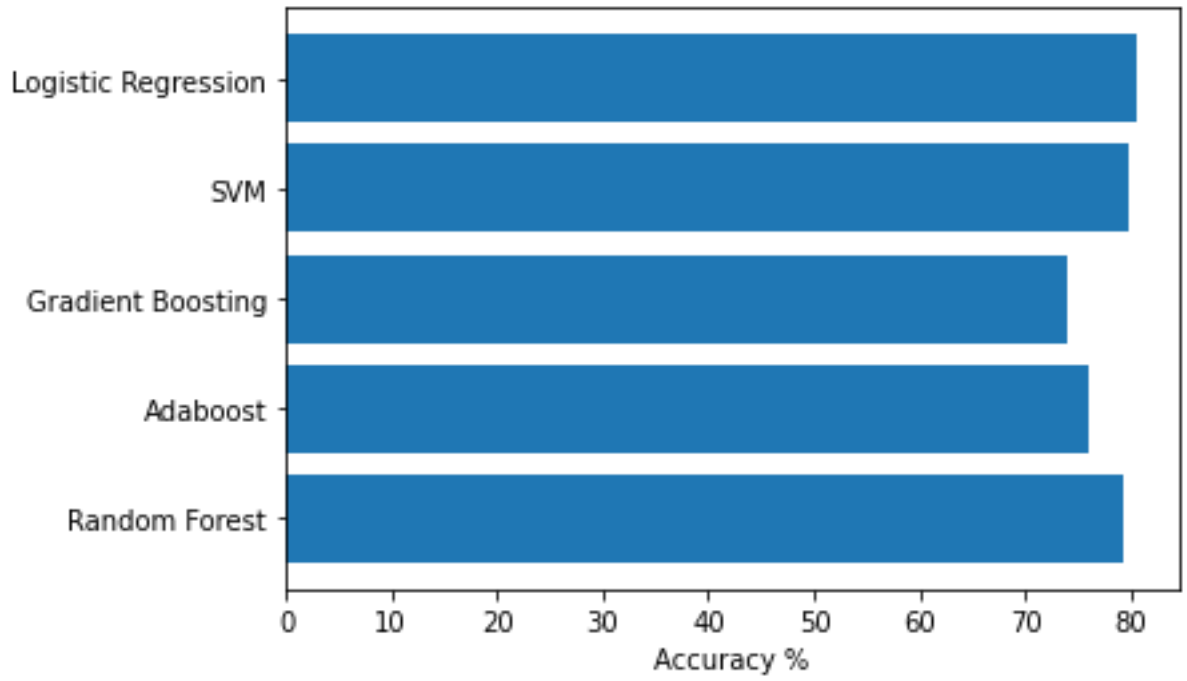

Figure 1: Accuracy (in %) for each model

Remember what was said earlier about precision? The author did not test for precision, but this study did for all five models. The results are as follows:

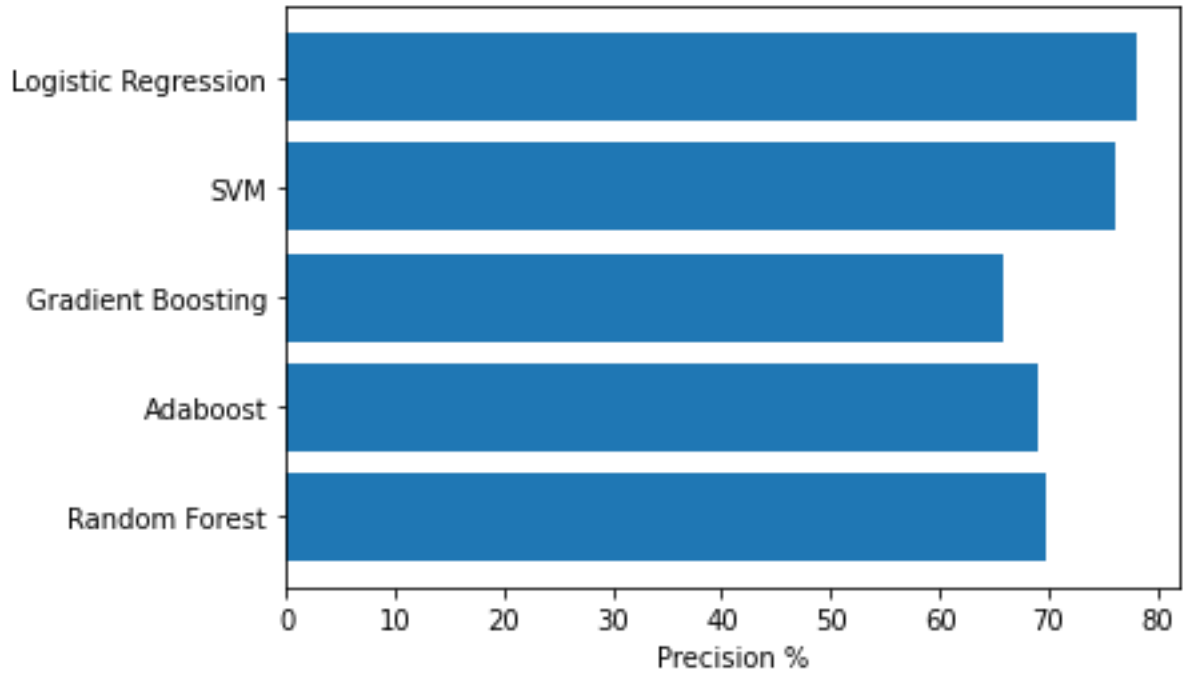

Figure 2: Precision (in %) for all models

Logistic regression still ended up being the best model. Thus, logistic regression was chosen as the best model overall. To double-check the precision values, the confusion matrix is below:

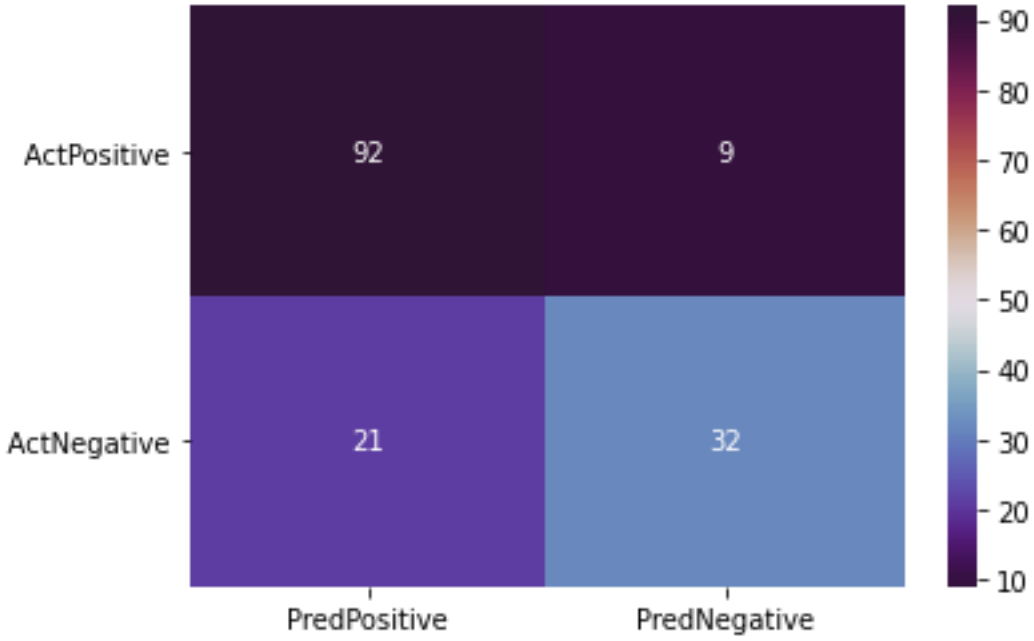

Figure 3: Confusion matrix for logistic regression

The most important things to note on the confusion matrix here are the dark squares. 92 is the number of patients that were correctly labeled as diabetic. It is desirable to have high values in this square, known as true positives. 9 is the number of patients that got a false negative test, meaning that they are actually diabetic but tested negative. This number should be low, as the equation for precision here is: 92/(92+9)

Conclusions and Recommendations

Logistic regression is technically the best model among the five for this study. It requires no hyperparameter tuning and scores highly in accuracy and precision. SVM is a close second with 1-2% change in accuracy and precision compared to logistic regression. Random forests

have a tendency to overfit on the training data, which is what most likely occurred here. Adaboost and gradient boosting work for some models, but this is a case where the simpler model tends to fit better, given that specific, cleaned data. Rather than spending several days tuning hyperparameters, it would be a better use of time to find out from doctors what values seem unusual for those with diabetes, and then either removing those rows or imputing values. Also, it would be beneficial to find the original dataset and see which models work for having more data. As it is, this study is useful to determine who to test for diabetes among females over 21 who are of Pima Indian heritage. This study can also be expanded upon with other machine learning methods or deep learning methods. Most importantly, this study should be used as a lesson in proper data cleaning and looking at the bigger picture when deciding how much time to spend on a section of the study. Clean data makes for a quicker modeling process, and no amount of tuning bad data will fix the problem.