Alexia Marcous

Springboard Data Science Career Track

August 5th 2019 Cohort

# Capstone Project 1 – Predicting Hypertension

# In-depth Analysis with Machine Learning

## Choosing Algorithms

Algorithms were chosen based on the properties of the data and target variable. The data from the National Health Interview Survey and is labeled and contains the target variable, hypertension, making this a supervised learning problem. The target variable has two categories, yes/no, making this a binary classification problem. Five algorithms were chosen for this problem type, including K-Nearest Neighbors (KNN), Logistic Regression, Random Forest Classifier, Adaptive Boosting (AdaBoost) and Extreme Gradient Boosting (XGBoost).

## Choosing Evaluation Metrics

Evaluation metrics were chosen based on the properties of the data and the implication of false predictions. In this case, the data is imbalanced, with 39% of responses positive for hypertension. For this reason, F1 score was chosen over accuracy.

The F1 score is also the appropriate evaluation metric due to a difference in the cost of false positives (predicting hypertension when the condition does not exist) and false negatives (predicting no hypertension when the condition does exist), making it better to focus on the true positive rate, also known as sensitivity or recall, and precision, or how "precise" the classifier is when predicting positive instances. Since the F1 score is the weighted average of precision and recall, if the F1 score is high, both precision and recall of the classifier indicate good results.

## Initial Results

The first algorim run was KNN, one of the simplest algorithms for classification tasks, with no parameters to fit. The next algorithm run was logistic regression, where the max iterations parameter was increased until a convergence warning was resolved at a value of 4,000. The third algorithm run was Random Forest Classifier, with default parameters.

|  |  |  |
| --- | --- | --- |
|  | Positive F1 | Negative F1 |
| KNN | 0.38 | 0.82 |
| Logistic Regression | 0.45 | 0.84 |
| Random Forest | 0.49 | 0.83 |
| AdaBoost | 0.47 | 0.88 |

## Improving Results – Additional Data Wrangling and Feature Selection

The first approach at improving these results was to review the data again. Based on re-examining usable data, one feature was dropped and missing data for two features was removed, resulting in a dataset of 199, 569 rows and 39 columns. Feature importances were reviewed but were inconsistent among models. The models were re-run on this set, showing improvements on the Positive F1 scores:

|  |  |  |
| --- | --- | --- |
|  | Positive F1 | Negative F1 |
| KNN | 0.43 +11% | 0.74 |
| Logistic Regression | 0.53 +17% | 0.78 |
| Random Forest | 0.54 +10% | 0.84 |
| AdaBoost | 0.56 +19% | 0.78 |

The second approach at improving these results was to test the impact of scaling the data. The scaling preprocessing method was tested with KNN, improving the Positive F1 score to 0.47. Since this result was not as good as other models, scaling was not implemented.

## Improving Results – Tuning Algorithms, Additional Training/Testing

The third approach at improving these results was to adjust the settings on the two top-performing algorithms and run them through more training and testing iterations. This was achieved with hyperparameter tuning and cross validation via grid search and random search methods. Multiple rounds of different values for parameters were perfomed, until the results leveled off here:

|  |  |  |
| --- | --- | --- |
|  | Positive F1 | Negative F1 |
| Logistic Regression | 0.54 +1% | 0.78 |
| AdaBoost | 0.58 +3% | 0.79 |

## Final Results - XGBoost

At this point, these results seemed to be about the best we could get with these models, and it was time to try XGBoost, the algorithm that consistently outperforms other models. Using multiple rounds of hyperparameter tuning and cross validation, XGBoost produced the highest Positive F1 score and next-highest negative F1 score:

|  |  |  |
| --- | --- | --- |
|  | Positive F1 | Negative F1 |
| XGBoost | 0.60 | 0.78 |

XGBoost also produced a model with the highest precision, recall and specificity.

* If the model predicts hypertension, this result will be correct 68% of the time (precision)
* If the subject has hypertension, the model predicts this accurately 54% of the time (sensitivity, recall or true positive rate)
* If the subject does not have hypertension, the model predicts this accurately 83% of the time (specificity or true negative rate)

## Summary

With the current data available, this model is specific and precise but not sensitive. Reasons for the lack of sensitivity could be due to the nature of survey data – not all respondents answer all questions, and not all questions are answered are answered accurately.

Fortunately, despite the low sensitivity, the model can still provide benefit. Tests with high specificity provide value when the result is positive, as they can be used for ruling in patients who have a certain disease. With the current results, the usefulness of the model is based less on its ability to predict hypertension, but more on its function as a screening tool for identifying people at risk for hypertension. This model could be used in community outreach efforts to raise awareness about hypertension and identify people who should follow up with their primary care provider to have their blood pressure tested. It could also be used to flag patients in the healthcare setting for a focused assessment of hypertension.

## Future Improvements

These results could be potentially be improved with additional efforts on feature selection, such as using methods designed for this purpose available in Scikit-Learn (the SelectKBest method) and pruning features based on their importance in tree-based algorithms.