Predicting the Diagnosis of Treatment

Maxwell Lee

Viterbi School of Engineering, University of Southern California,

Los Angeles, California 90089, USA

(Dated: January 19, 2022)

Abstract

In this report, we will detail the findings of a simple logistic regression model used to predict the classification of patient's treatment. We are given traditional historical medical data as well as the results of tests and genetic analysis for each patient, as well as the diagnosis of treatment or not. We are interested in creating a model that is as accurate as possible, while also factoring the cost of the more extensive testing. We were able to create a model with an AUC score of .735 on training data, with a similar accuracy rate of 74.2%. We found that TestA, GeneC, GeneD, and GeneF provided extremely minimal beneficial information to our model. TestB and GeneE provided more benefit than those previously mentioned, although even this benefit is very small. As these tests are expensive and difficult, it makes sense to only include TestB and GeneE or even just the traditional data if we were to use this model in the future. The most important features are gender, age, blood pressure, and family history. The most likely candidate for treatment would be a younger girl with high blood pressure and a family history for the disease.

I. INTRODUCTION

For this task, we are given medical data about a patient and are asked to predict whether or not a treatment was prescribed to the patient. The medical data can be considered of two varieties: basic and advanced. The basic features we are given are the age, blood pressure, gender, a blood test, and the family history. All of this information is easy and inexpensive to obtain. The advanced information is TestA, TestB, GeneC, GeneD, GeneE, and GeneF. These tests are more expensive and difficult, so we are especially interested in their relevance to our model. If we can create a reasonably accurate without their inclusion, that would make the deployment of this model far more practical.

To evaluate our model, we will use AUC score as a singular way to compare models and make decisions regarding feature inclusion or hyperparameter tuning. This is an all encompassing evaluative tool of a classifier, and by only having one metric we can make more declarative statements about the relative performance of models. However, other classification metrics such as accuracy, F1 score, precision, recall, and confusion matrices will be used to provide greater context to the performance of the final model.

II. DATA EXPLORATION

To explore the data, it was read into a Jupyter Notebook using a Pandas DataFrame. The DataFrame has 12 columns and 7500 rows, with each column representing a feature and a row representing a patient. The data was initially examined for missing values. The attribute for family history is missing 2607 times, or roughly in a third of the rows. We will detail how this is remedied in Section III.

The next step was creating boxplots for each of the continuous features, being age, blood pressure, TestA, and TestB. Each of these features have a relatively normal distribution, which is useful for a logistic regression model.

We then explored the categorical data by creating histograms for each categorical feature. We start with the treatment variable, which tells us that treatment is recommended roughly 55% of the time. This can be considered an extremely rough baseline for our model. To be confident there is signal in our data, we should have a model that is accurate more than 55% of the time, as we could achieve that accuracy by simply recommending treatment every

time.

There are other interesting findings in some of the categorical variables. Family history has a true value in extremely few circumstances, with the vast majority of values being either false or null. Blood tests have a similar trend, with the majority of blood tests being negative. This doesn't necessarily rule out their usefulness. While family history has a very low frequency of true, when family history is true treatment is recommended about 93% of the time. So while a false or missing value may not provide much information, a true value can immediately point to treatment.

Finally, to understand some of the relationships that may occur in the logistic regression model, a simple correlation heatmap was created. The first notable correlation in the data is between age and TestA. At .969, age and TestA provide almost identical information. This is our first clue that TestA may not be useful to our model, as it may just be an expensive way to guess someone's age. No other features are particularly notably correlated to each other. As for correlations to our target variable of treatment, we can hypothesize that age and gender will be extremely relevant to our model, as they are both highly correlated with treatment.

III. DATA PREPROCESSING

The first step in Data Preprocessing was to detect outliers in the data. This was done by examining the boxplots created in Section II. Only blood pressure had values worthy of exclusion. 5 rows had blood pressures of -999, which is indicative of blood pressure being a missing value. Because this represents such a small percentage of the overall dataset, these values were dropped from the dataset and only positive blood pressures were considered. All other categorical features were reasonable values that were not necessary to exclude.

The next decision was how to handle the missing values in family history. Originally, these values are handled as three classes in a categorical variable, so as not to remove nearly a third of the rows in the dataset. However, the null class ultimately provided so little information that it was essentially treated the same as false. This is fine, as the vast majority of family history values are false.

To handle categorical variables, we employed One Hot Encoding using the $get_dummies()$ function in Pandas. We drop one of the columns created for each of the categorical features,

as including all would be redundant and have perfect colinearity.

Finally, we decided to scale the data using Sci-kit Learn's StandardScaler function [1]. This essentially transforms each column from its raw form into each value's z-score relative to itself. This is done so that each feature has a similar scale, making the interpretation of coefficient far simpler. This is done without loss in model performance, as AUC score improved very slightly with the scaling of data, from .739 to .744.

IV. MODEL SELECTION

We are limited to Logistic Regression for this classification problem as it is simple and easily interpretable. Therefore, we do not have much to do in the way of model selection. The most important decisions are in reference to which features should be included, which is discussed in length in Section VI. For this section, we have removed TestA, GeneC, GeneD, GeneF, and the na class of family history from the feature set.

While the main task is feature selection, there are some hyperparameters in Sci-Kit Learn's logistic regresssion model that can optimize performance [1]. To choose the best hyperparameters to use, we will use a cross validation grid search. The data is first split into an 80-20 split of training and testing data. 5-fold cross validation is then used on the training data, where each set of hyperparameters is tried and the average AUC score is computed for the validation sets. The hyperparameters with the best AUC score is then used as the best model. The hyperparameters tried are provided in Table I. The best parameters are penalty: l2, C: 0.1, max_iter: 100, and solver: saga.

Hyperparameters	Description	Values tried
Penalty	Penalty function applied to regularization	l1, l2, elasticnet
C	Inverse of regularization strength	.1, .5, 1, 5, 10
solver	Algorithm used for optimization	newton-cg, lbgfs, liblinear, sag, saga
max_iter	Number of iterations before optimization stopped	100, 500, 1000

TABLE I. Hyperarameter grid used for cross validation grid search

V. MODEL EVALUATION

Having decided on the best hyperparameters, we then retrain the model with the full training data, then see its performance on the data set aside for testing. The AUC score on the testing data is .743, F1 score is .773, and accuracy is 74.1%.

Of the 1499 observations in the test set, 808 were classified as positive, indicating a treatment diagnosis, while 691 were diagnosed as negative. In the prediction set, 897 were predicted as positive while 602 were predicted as negative. The precision for each class is similar at 73% and 75% respectively, while the recall for the negative class was considerably worse at 66% while the positive class's precision is 82%. This is indicative of a high false positive rate, which makes sense as 16% of total predictions were false positives.

A higher false positive rate may be preferable given the circumstances of the problem. If the treatment is highly invasive for a relatively low risk ailment, a higher false positive rate is obviously undesirable. But if the cost of not treating the ailment far outweighs the cost of the procedure for a person not needing the treatment, we may want a higher false positive rate. As we are not supplied the treatment, these are given equal weight, but if one is far more costly than another it is relatively simple to adjust the weights of our model to account for this fact.

VI. FEATURE IMPORTANCE

We can now the relative importance of each feature to our model. We will start by looking for features we can safely exclude from our model. We start by creating a model with all features as our baseline, then removing each feature individually to see how the AUC score is affected by its exclusion. The baseline model with all features has an AUC score of .746. Looking only at our advanced testing features as candidates for removal, we see that model performance is unaffected by the exclusion of TestA, GeneC, GeneD, and GeneF. For all of these features, their exclusion does not change the AUC score by even .01. TestB and GeneE's exclusions do drop the AUC score by about .01, which is relatively very small, but more noticeable than the previous features.

This process is repeated, except by adding features to a baseline model of only the traditional medical data features. This baseline model has an AUC score of .730, which

should be noted is only .016 less than the model with all features. We again see that by adding TestA, GeneC, GeneD, and GeneF the AUC score improves by less than .01, while the inclusion of TestB and GeneE improves the AUC score by about .01.

Finally, as we used the StandardScaler function, we can look at the coefficients as a guide for feature exclusion. Because these are all the same scale, we can take the absolute value of their coefficient as a measure of relative importance. Doing so confirms our findings that TestA, GeneC, GeneD, and GeneF, along with a null value of family history have almost no importance to our model. TestB and GeneE have some importance to our model, so if we really value that extra 1% their inclusion is valid. However, if their inclusion dramatically increases the cost of implementing this solution, they can be excluded with minimal cost to performance.

Having removed the features with no information, we are left with the following features: age, blood pressure, gender, blood test, family history, TestB and GeneE.

VII. INTERPRETATION

In deciding to use the StandardScaler, we increased our ability to compare relative importance of features, but lost a bit of our ability to easily interpret our model. Even without scaling the data, the coefficients are not as simple to interpret as a linear regression model. Therefore, we will call out some of the directions of the important relationships between the features and treatment.

The most important feature is gender. A female is far more likely to receive treatment than a non-female. The likelihood of treatment decreases with age, meaning younger people are more likely to receive the treatment. A higher blood pressure increases the likelihood of treatment, while a positive blood test slightly decreases the likelihood of treatment. A family history of true also increases the likelihood rather significantly, as hypothesized in Section II. Finally, an increase in the TestB results and possesing GeneE both decrease the likelihood of treatment. A plot showing the coefficient values used to make these determination values is showing in Figure 1.

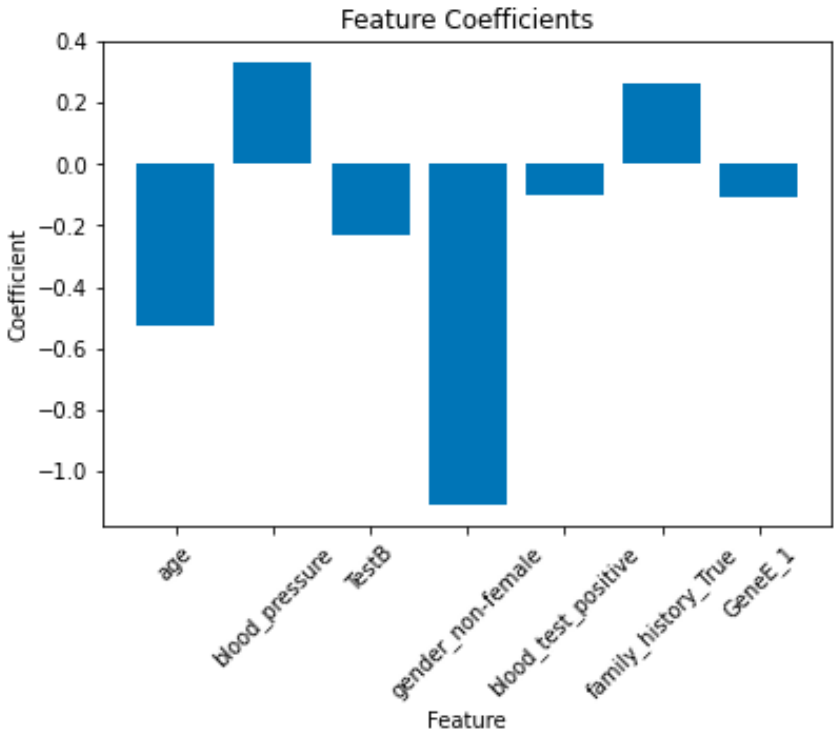

FIG. 1. Coefficients for final model

VIII. CONCLUSIONS

With the given features, we are able to create a simple model with decent performance. More sophisticated modelling techniques may be able to extract better performance at the expense of interpretability. Importantly to our task, we were able to show that the benefits of including the advanced testing we minimal at best. Only TestB and GeneE provided any gain in performance for our model, and even their inclusion only improved the AUC score from .730 to .735 on testing data. Given the cost of these advanced tests, it may be in our best interest to simply use the traditional medical data, at least until there is advanced testing more useful to us, or the cost of existing testing is dramatically reduced.

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research **12**, 2825 (2011).