Predicting Health Problems with Nutritional Status

Ludan Zhang

Jiachen Zhang

Yue Peng

School of Public Health

Department of Biostatistics

Brown University

Abstract

Using data from National Health and Nutrition Examination Survey, this study conducted linear regression to fit model for blood pressure and used multiple techniques including logistic regression, feature selection, CatBoost, XGBoost, LightGBM and Deep Neutral Network to fit models for diabetes. Linear regression using L_1 regularization(Lasso) selected weight as the explanatory variable for blood pressure and gave best prediction, comparing with L_2 regularization (Ridge) and Elastic Net methods. As for diabetes, CatBoost won the single model performance. Feature selection didn't improve the model's performance, but feature engineering did improve the performance. CatBoost with CNN feature engineering gave the best prediction with area under the curve equals 0.7583.

Keywords: Linear Regression, Logistic Regression, Feature Selection, CatBoost, XGBoost, LightGBM, Deep Neutral Network, Convolutional Neural Network

Introduction

Background

The data used in this project came from the National Health and Nutrition Examination Survey (NHANES). NHANES is a program of studies designed to assess the health and nutritional status of adults and children in the United States. The survey is unique in that it combines interviews and physical examinations. The NHANES interview includes demographic, socioeconomic, dietary, and health-related questions. The examination component consists of medical, dental, and physiological measurements, as well as laboratory tests administered by highly trained medical personnel.

Objectives

This project is aiming to find the most influential factors for blood pressure and diabetes; build suitable models to predict blood pressure and diabetes; and access models' predict accuracy.

Method

Datasets and variables

Downloaded seven datasets from NHANES website[1]. Data from 2001 to 2014 were used in this project. Alcohol dataset includes alcohol consuming information. The blood pressure dataset includes four measurements of systolic blood pressure and four measurements of diastolic blood pressure. Body measurements dataset contains variables: height, weight, and BMI.

[1] https://www.cdc.gov/nchs/nhanes/index.htm

The demographic dataset contains variable: age, gender, education level, and marital status. The two nutrition datasets contain participants' self-reported information about the amount of protein, fiber, carbohydrate, vitamins, water and other necessary nutrition that were taken during the two sampling day. The cardiovascular health dataset includes information about participants cardiovascular health.

Data preprocessing

Since the 7 datasets has different information and subjects, we selected the subjects(n= 24424) who have records in all the 7 datasets first. Then we combined the 7 datasets based on the subject selected using the subject ID. After the combination, we deleted the subjects who have missing values for outcomes of interest. Furthermore, according to the documentation, we replaced some original values by more meaningful values. For example, 99 and 77 represent don't know and refuse to answer for some features, so we replace 99 and 77 by NAs(missing values). Finally, we split the dataset into train and test sets. The first 7 years' data were used as train data(n=19159) and the 8th year's data consist the test set(n=3505).

After creating the train and test set, we did some simple feature selection based on the train set so that we could have a more reasonable dataset to train the models. We examined the number of missing values each feature had and delete the features which have more than 30% missing values. This step left us 41 features for Diabetes, and 38 features for blood pressure.

Model fitting

We used several methods to fit the models. Since blood pressure is a continuous variable and following normal distribution, we use linear regression to fit the models.

Linear Regression:

We used R to conduct linear regression on blood pressure using L_1 regularization (Lasso), L_2 regularization (Ridge) and Elastic Net methods. We used cross validation with 10 folds to select the parameter λ that controls the amount of shrinkage.

To fit the models for diabetes, we used the following methods:

Logistic Regression:

We used R to fit the logistic regression. Also, we used logistic regression with L1 regularization (mentioned as LASSO in the following parts) to do feature selection. Since LASSO in R cannot handle any missing values, we filled all the missing values by 0.

Random Forest:

We used R to fit the Random Forest model as well. We filled NA with median. The number of trees was set as 500 after doing cross validation with 10 folds. And the trees are using all the input features when fitting.

CatBoost:

One advantage of CatBoost is that the author makes a hypothesis of the difference between the actual sampled sample and the real sample, and believes that their feature distribution is not completely consistent. This kind of assumption is in fact very consistent with the actual situation encountered, especially if the sample size is not sufficient. Catboost is more stable on small sample sets than LightGBM and XGBoost. After filling NA with median, we train 400 iterations, and extracted feature scores from it since it is a tree-based model.

XGBoost:

Boosting classifier belongs to ensemble models, the basic idea is to aggregate hundreds of less accurate tree-based models to form a very accurate model. This model usually iteratively generates a new tree-based model at each step. XGBoost is a gradient boosting implementation

which has been widely used. We used python to fit the XGBoost model as well. We filled NA with median. The max depth was set as 6 after doing cross validation with 5 folds. And the trees are using all the input features when fitting.

LightGBM:

Released from Microsoft, LightGBM has been claimed to be more efficient (better predictive performance for the same running time) than XGBoost. After filling NA with median, we train 3000 iterations.

Deep Neural Network:

We used four hidden layers neural network to optimize the log-loss. After the 10 epochs, the auc of training set was 51%, which was exactly not appropriate to use for this dataset. Also, the lack of explainability was the cons of deep learning model.

Feature selection

To do feature selection, we used two different approaches.

Feature selection with LASSO:

We used cross validation with 10 folds to select the best penalty for LASSO. Then, we did the following process:

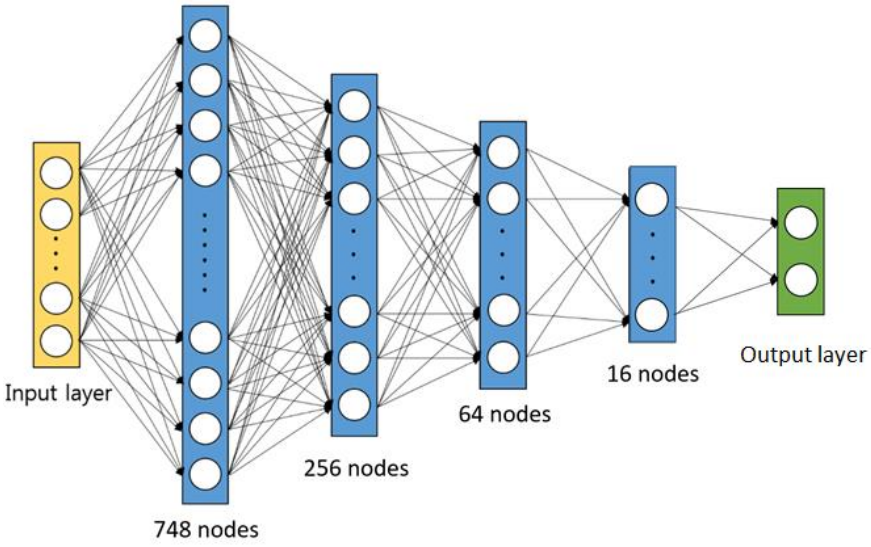
a. Randomly choose 80% observations from the train set to form a new train set.
b. Fit the LASSO with this new train set, and record the coefficients of the features.
c. Give 1 score for those features whose coefficients are not zero, 0 score for others.
d. Do a to c for 100 rounds, add the scores of all rounds.
e. Select the features whose scores are not zero.

Feature selection with CatBoost:

We chose the top 18 features had highest feature scores by using *get_feature_importance* parameters in CatBoost Model.

Feature engineering

Feature engineering with DNN



To directly use this four layers neural network model was not good, but we brought up a thought that we could use some outputs from hidden layers as new feature vector.

Feature engineering with CNN

1. Pick 18 of the most important features, concatenate them into a (sample size, 36) matrix.
2. Concatenate this matrix with the first full-connected layer in DNN. So far, we obtain a (sample size, 784) matrix.
3. Reshape the features as a (28 * 28) “image” for each sample.
4. Input (sample size, 1, 28, 28) to LeNet Network
5. After training, we output the feature map of conv_1 layer.
6. Since the important features generated by CatBoost lie in the first 36 columns, the information will be stored within columns[0, 20] after a max-pooling layers.

7. Add these 20 features into CatBoost Model, then train.

Result

Blood pressure

We conduct linear regression on blood pressure. The response variable is the average blood pressure, calculated by: $\frac{1}{3} systolic\ pressure + \frac{2}{3} diastolic\ pressure$. There were 38 explanatory variables.

Method	MSE
Ridge	0.5462173
Lasso	0.1356925
Elastic Net	0.1362492

Table 1. Mean Squared Error for Each Methods

The Lasso method gave the best prediction, with the smallest mean squared error. Lasso method only selected one explanatory variable, weight, out of the 38. Thus, the final model for blood pressure is as follows:

Average Blood Pressure= 48.057+5.751*Weight(kg)

Diabetes

Models with all the features

First of all, we tried to use the methods to fit model with all the 39 features after deleting 2 features from the 41 features selected in the data preprocessing step. One feature we deleted was *agemonth(age computed in months)* which was highly correlated with *ageyear(age*

computed in years) but have some missing values. The other feature deleted was *blood pressure* which was computed by the *diastolic pressure* and *systolic pressure*.

We used the AUC(area under curve) on test set to evaluate our model, table 2 shows the performance of different models. We can tell that CatBoost, LigtGBM and Logistic Regression can produce relatively better results.

Method	CatBoost	LightGBM +CatBoost	LightGBM	Logistic Regression	XGBoost	Random Forest	DNN
AUC	0.7527	0.7475	0.7471	0.7461	0.7426	0.7226	0.5179

Table 2. Results of models with all the features

Models after features selection

We selected 21 features by LASSO, table 3 shows the order of these features, the order can tell the relative importance of these features. Then, we used logistic regression to fit model with these 21 features, the AUC on test set is a little lower than the AUC of the model with 39 features, which means that the feature selection didn't improve the model's performance.

We also select 18 features by CatBoost, table 3 shows the order of these features, the order can tell the relative importance of these features. Then, we used CatBoost again to fit model with these 18 features, the AUC on test set is a little lower than the AUC of the model with 39 features, which means that the feature selection didn't improve the model's performance.

Method	Features Selected	AUC
LASSO + Logistic Regression	ALQ100, ALQ150, Year, Ageyear, Oldedu, Avgsymeasure, Waist, Sugar, Fiber, Alcohol, Plainwater, Avgdimeasure, Chol, Gender, Bmi, Phos, Iron, vitC, Calcium, vitB1, caff	0.7448
CatBoost	Waist, sugar, ageyear, avgdimeasure, alcohol, avgsymeasure, oldedu, bmi, fiber, weight, year, tapwater, height, carbo, caff, plainwater, energy, vitC	0.7511

Table 3. Results of models after feature selection

Models after feature engineering

We used the AUC(area under curve) on test set to evaluate our model after feature engineering, table 3 shows the performance of different models we tested. We can tell that CatBoost with DNN features or CNN features can produce relatively better results. And adding features from convolutional layer did best among all models we tried.

Method	catboost+dnn_fc3(64)	catboost+dnn_fc4(16)	Catboost+cnn_conv1[:, 0:20]
AUC	0.7549	0.7558	0.7583

Table 4. Results of models after feature engineering

Discussion

In the feature selection process, the original features are transformed in multiple layers by means of DNN or CNN, and the original features are mapped into a new space to improve the classification ability, but at the same time, the interpretability of the model is reduced.

While using top 18 features to fit in LightGBM model, we hit a highest training AUC

0.96, but there was no increasing of test AUC. We considered this was an over-fitting situation. Thus, we would not show the result of this case.

Using Youdan's Index to find optimal cut-off point in our best model (CatBoost with convolutional layer features), we obtain the threshold is 0.17. The sensitivity is 0.7355, and the specificity is 0.6583.

References

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction*.