**MScBMI 33200 – Machine Learning for Biomedical Informatics**

**Assignment I**

**<Troy Zhongyi Zhang>**

**Directions:**

1. Fill out below information (tables and methods)
2. Write as much detail as possible for the methods section for each problem. The idea is for you to have a record of all the steps you took for building each model.
3. Submit this document along with your code in an HTML/PDF format

**Section 1: EMR Bots 30-day Readmission study**

**Using the training datasets, create the following models:**

**1. Naïve model: This model utilizes only patient characteristics (age, gender and race) to predict 30-day readmission in a logistic regression framework**

**2. Logistic Regression model : This model utilizes patient characteristics and most-recent lab recordings to predict 30-day admissions in a logistic regression framework.**

**Calculate AUC on the test dataset. Fill out the following Table.**

|  |  |
| --- | --- |
|  | AUC (95% CI) |
| Naïve Model | 0.4909 (0.4105, 0.5714) |
| Logistic Regression | 0.4843 (0.3980, 0.5706) |

**Insert details (packages, parameter selection, etc.) on the models that were developed in the space given below.**

**Methods:**

Firstly I transferred all the texts data to categorical data for binary classification, and then I selected the training set (year<=2004) and testing set (year>2004). Since the Naïve Bayes model and logistic regression used different features, I have to prepare two data sets for modeling.

A close up of text on a white background

Description automatically generated --> Naïve

For logistic regression:

A screenshot of a cell phone

Description automatically generated

I used Python for modeling. I imported GaussianNB for the Naïve Model and LogisticRegression for the Logistic Rregression model from sklearn package.

from sklearn.naive\_bayes import GaussianNB

I fitted the training set and predicted testing set

Then I used predict\_proba to calculate the probability of my predictions.

nb – Naïve model: lg – logistic regression:

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Both Naïve Model and Logistic Regression give me the same prediction results. All the predictions for testing set are “0”. I imported metrics and classification\_report from “sklearn” packages for visualization on the accuracy scores by confusion matrix:

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

By using the roc\_auc\_score from sklearn package, I got 49.09% for AUC of Naïve Model

A screenshot of a cell phone

Description automatically generated

I got 48.43% for AUC of Logistic Regression Model

A screenshot of a cell phone

Description automatically generated

Finally, since Python doesn’t have the pROC package to calculate the confidence interval, I found two ways to obtain the 95% CI.

The first way is bootstrapping to draw the ROC curve and compute the CI, and the second way is to use the DeLong function transplanted from the pROC package in R.

Naïve: Bootstrapping:

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Naïve for transplanted pROC package: Logistic Regression for pROC DeLong function:

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Since bootstrapping method has a lot of uncertainties, I would choose the transplanted pROC package with DeLong function for my CI computation. The Confidence Intervals I filled in above form are the pROC calculation results.

**Section 2: Gusto Study**

**Using the training datasets, create the following models:**

1. **GLM model : This model utilizes all features to predict 30-day mortality in a logistic regression framework.**
2. **Ridge Regression model : This model utilizes all features to predict 30-day mortality in a logistic regression framework with regularization. Utilize a 5 fold cross validation to build the parameters for your model.**

**Calculate AUC on the test dataset. Fill out the following Table.**

|  |  |
| --- | --- |
|  | AUC (95% CI) |
| Logistic Regression | 0.8285 (0.7919, 0.8650) |
| Ridge Regression | 0.8280 (0.7912, 0.8647) |

**Insert details on the models that were developed in the space given below.**

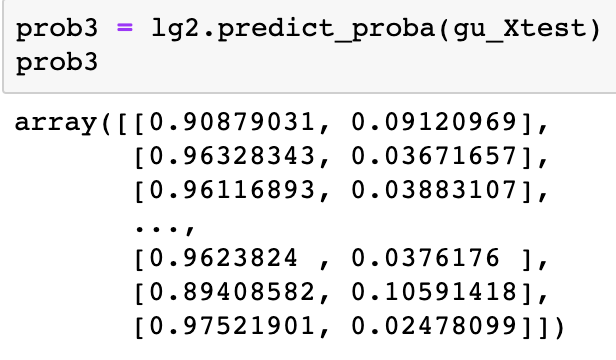
**Methods:**

Firstly I transferred the text data into categorical int data for classification and split training dataset (GROUP = ‘sample2’, ‘sample4’, and ‘sample5’) and testing dataset(GROUP = ‘west’).

A screenshot of a cell phone

Description automatically generated

Then I used LogisticRegression package from sklearn package to model the data for predictions. Below is the probability of predictions



Then I I imported metrics and classification\_report from “sklearn” packages for visualization on the accuracy scores by confusion matrix for logistic regression:

A screenshot of a cell phone

Description automatically generated

By using the roc\_auc\_score from “sklearn” package, I got 82.85% for AUC of logistic regression for GUSTO dataset:

A screenshot of a cell phone

Description automatically generated

I import the RidgeCV from the “sklearn.linear\_model” package for regularization. I chose some alphas values to try some different parameters. Then I set “cv = 5” in the parenthesis according to the question requirement. By using the roc\_auc\_score from “sklearn” package, I got 82.80% for AUC of logistic regression for GUSTO dataset:

A screenshot of a cell phone

Description automatically generated

For the 95% confidence intervals, I also did both ways (bootstrapping and pROC) to compute the intervals for logistic regression and Ridge on the GUSTO dataset. I believe the pROC will give a better and more accurate result, I will just show the details of the pROC DeLong computation results.

For Logistic Regression on GUSTO dataset, I got the 95% CI of (0.7919, 0.8650).

A screenshot of a cell phone

Description automatically generated

For RidgeCV with 5 cross-validations on GUSTO dataset, I got 95% CI of (0.7912, 0.8647)

A screenshot of a cell phone

Description automatically generated

**Section 3: Short Answer Questions**

Please answer the following questions briefly.

**a. What are the assumptions of a linear prediction model?**

Ans.

The regression has five key assumptions:

1. Linear relationship.
2. Multivariate normality.
3. No or little multicollinearity.
4. No auto-correlation
5. Homoscedasticity

Linear regression needs the relationship between the independent and dependent variables to be linear. The linear regression analysis requires all variables to be multivariate normal. Linear regression analysis requires that there is little or no autocorrelation in the data.

**b. Explain briefly what the interpretation of R2 is.**

Ans.

R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. Whereas correlation explains the strength of the relationship between an independent and dependent variable, R-squared explains to what extent the variance of one variable explains the variance of the second variable. So, if the R2 of a model is 0.50, then approximately half of the observed variation can be explained by the model's inputs.

**c. What is learning rate in the gradient descent algorithm? Explain what happens when it is too high. Explain what happened when it is too low.**

Ans.

The learning rate is the step size on how much the gradient descent model moves toward the global minimum. If the learning rate is too high, it will possibly pass over the minimum point. If the learning rate is too low, it will take the model too long to reach to the minimum point.

**d. Consider the following scenario. You developed two models for predicting mortality in the hospital using the same dataset. Model 1 had an AUC of 0.78 (95%CI 0.76-0.80). Model 2 had an AUC of 0.72 (95% CI 0.70-0.73). When you took a closer look at the sensitivity and specificity measures for various thresholds (predicted probabilities x 1000), you saw the following:**

|  |  |  |
| --- | --- | --- |
| Model Cutoff | Sensitivity (%, 95%CI) | Specificity (%, 95% CI) |
| Model 1 (Predicted Probability x1000) | | |
| ≥ 17 | 57 (55-59) | 88 (87-88) |
| ≥ 20 | 53 (51-55) | 90 (90-91) |
| ≥ 23 | 49 (47-51) | 92 (92-92) |
| ≥ 27 | 40 (38-41) | 93 (93-93) |
| ≥ 67 | 19 (18-21) | 99 (99-99) |
| Model 2 (Predicted Probability x 1000) | | |
| ≥ 16 | 57 (55-59) | 90 (90-91) |
| ≥ 20 | 40 (38-41) | 97 (97-97) |
| ≥ 30 | 21 (19-22) | 99 (99-99) |

**Which is a better model to operationalize? And why?**

Ans. The better model to operationalize is really depends, but I would like to choose Model 1 to be a better model.

A close up of a map

Description automatically generated

By magnifying the plot through modifying the x-axis and y-axis, I got a plot like

A close up of a map

Description automatically generated

Although Model 2 (red line) is above the model 1 (blue line), the Model 1 has higher AUC\_ROC than model2’s AUC\_ROC. If the data is too specific, the dataset is more likely overfitting. However, Model 1 has higher AUC compared with model 2, which means comprehensively speaking, the Model 1 could predict the True Positive more accurately. The two models have very similar sensitivity, but model 1 has higher specificity, which means model 1 could predict the True False more accurately. I would pick Model 1 to be a better model to operationalize.

**e. Consider the following calibration plot. What can you infer from this plot?**



Ans.

The y-axis is the relative frequency of what was observed, and the x-axis is the predicted probability frequency. Usually the closer the points appeared along the main diagonal from bottom left to upper right, the better calibrated or more reliable a forecast is. The line usually over-forecasts low probabilities and under-forecasts high probabilities. As the predicted probability of outcome increases, the outcome prevalence also increases. The plot line is always above the diagonal means the model has under-forecast and the probabilities are too small.

Section 4 (Optional): Watch the following talk: Yann Le Cunn: Who’s afraid of non-convex loss functions? <https://www.youtube.com/watch?v=8zdo6cnCW2w>