Jon Rippe  
CSCE A415  
HW #2.2

## Problem 2: Logistic Regression

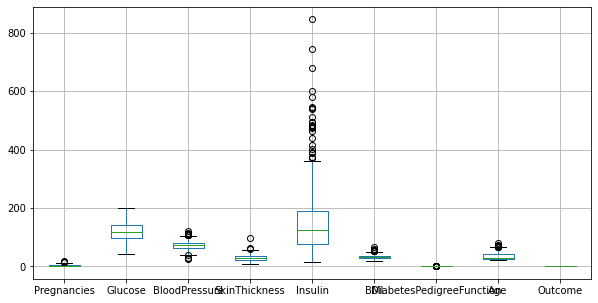
### Data Manipulation

I’ve experimented with three approaches:

1. Removing **all** missing information and highly correlated attributes.
2. Removing **all** missing information.
3. Using linear regression to fill in **some** missing information.

#### The following was implemented in all three approaches:

The dataset contains both *NULL* values and *0*’s where information is missing. The *0*’s have been converted to *NULL* for missing-data consistency (5).

Most of the missing information is in the *SkinThickness* and *Insulin* columns (6 - missingno). There are 44 rows that contain *NULL* values **outside** of those two columns (7). These rows compose about 5.7% of the dataset and will be dropped (11).

The *Insulin* column shows high variability (large number of outliers (9 - boxplot)) and a high correlation (0.58) with the *Glucose* column (10 - corrMatrix). Because of this, it is dropped (12).

#### Approaches #1 & #2:

The remaining missing data is limited to the *SkinThickness* column, which has a high correlation (0.65) with the *BMI* column (10 – corrMatrix). Because of this, it is also dropped (12).

Approach #1 drops the *Pregnancies* column due to its high correlation (0.54) with the *Age* column (10 – corrMatrix). I chose to drop *Pregnancies* instead of *Age* because *Age* has a higher correlation with *Outcome*.

#### Approach #3:

This approach uses linear regression to fill in the missing data in the *SkinThickness* column using the data in the *BMI* column, with which it shares a high correlation. This had the expected effect of increasing the correlation even more, which will probably have a negative impact on the final result.

### Testing (15):

Score Results

#1

0.7790

Train/Test Score

90/10 0.7940

80/20 0.7731

70/30 0.7748

60/40 0.7677

50/50 0.7752

40/60 0.7677

30/70 0.7639

20/80 0.7627

10/90 0.7554

#2

0.7762

Train/Test Score

90/10 0.7825

80/20 0.7756

70/30 0.7659

60/40 0.7679

50/50 0.7674

40/60 0.7695

30/70 0.7668

20/80 0.7630

10/90 0.7540

#3

0.7693

Train/Test Score

90/10 0.7756

80/20 0.7668

70/30 0.7662

60/40 0.7669

50/50 0.7652

40/60 0.7628

30/70 0.7623

20/80 0.7619

10/90 0.7474​

All three approaches were run through a logistic regression model (15). First, they were fitted and tested against their entire respective datasets (no train/test split) to see the overall accuracy of the logistic regression model. They were then tested using a range of train/test split ratios (50 iterations per split averaged together) to see how each affected the perceived accuracy of each model.

### Results (15):

Having fewer highly correlated attributes in the final model improves the model’s overall accuracy, even if only slightly as my models show (0.779 > 0.776 > 0.769). Additionally, the more valid data that’s used to train the logistic regression model, the more accurate the model becomes – as seen by the change in accuracy between different train/test ratios. However, using a high train/test ratio may inflate the model’s score due to the relatively small test pool. A formal proof would be needed, but I believe training a regression model using all rows of a dataset and then testing the model using the same data (as I did before doing the splits) is a better approach to gauging the overall accuracy of that model.

As a sidenote/afterthought, the linear regression models of this dataset were much better at classifying non-diabetic entries than they were at classifying diabetic entries, with a false positive rate of about 12% and a false negative rate of about 42% (15 – confMatrix).

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Predicted Class | |
|  |  | Non-Diabetic | Diabetic |
| Actual  Class | Non-Diabetic | 420 | 55 |
| Diabetic | 105 | 144 |

Confusion Matrix for an instance of Model #1