Homework 10

Jeff Tilton

10/27/2018

Question 14.1

The breast cancer data set breast-cancer-wisconsin.data.txt from http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/ (description at http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29) has missing values. 1. Use the mean/mode imputation method to impute values for the missing data. 2. Use regression to impute values for the missing data. 3. Use regression with perturbation to impute values for the missing data. 4. (Optional) Compare the results and quality of classification models (e.g., SVM, KNN) build using (1) the data sets from questions 1,2,3; (2) the data that remains after data points with missing values are removed; and (3) the data set when a binary variable is introduced to indicate missing values.

Goals

1. Build 5 data sets, 4 with different imputation techniques and 1 removing data points that have missing values
   - Mean
   - Mode
   - Regression
   - Regression with perterbation
2. Campare the results and quality of classification models with the 5 data sets

Data

```
##    Sample_code_number Clump_Thickness Uniformity_of_Cell_Size
## 1:            1000025               5                       1
## 2:            1002945               5                       4
## 3:            1015425               3                       1
## 4:            1016277               6                       8
## 5:            1017023               4                       1
## 6:            1017122               8                      10
##    Uniformity_of_Cell_Shape Marginal_Adhesion Single_Epithelial_Cell_Size
## 1:                        1                 1                           2
## 2:                        4                 5                           7
## 3:                        1                 1                           2
## 4:                        8                 1                           3
## 5:                        1                 3                           2
## 6:                       10                 8                           7
##    Bare_Nuclei Bland_Chromatin Normal_Nucleoli Mitoses Class
## 1:           1               3               1       1     0
## 2:          10               3               2       1     0
## 3:           2               3               1       1     0
## 4:           4               3               7       1     0
## 5:           1               3               1       1     0
## 6:          10               9               7       1     1
##          Sample_code_number             Clump_Thickness
##                           0                           0
##     Uniformity_of_Cell_Size    Uniformity_of_Cell_Shape
##                           0                           0
##           Marginal_Adhesion Single_Epithelial_Cell_Size
##                           0                           0
##                 Bare_Nuclei             Bland_Chromatin
##                          16                           0
##             Normal_Nucleoli                     Mitoses
##                           0                           0
##                       Class
##                           0
```

The data contains 699 points which means that no more than 35 values or 5% of any column should be imputed. There only seem to be 16 missing values located in a single column, Bare Nuclei, well under our 5% threshold.

Datasets

Method

Mean and Mode

1. Find the Bare Nuclei mean and mode
2. Create 2 copies of the original data set and apply the mean and mode to the missing data

Regression and Perturbation

This was much more complicated then the mean and mode. I chose to compare two types of regression, Elastic Net and Random Forest and choose the model with the lowest Mean Squared Error (mse).

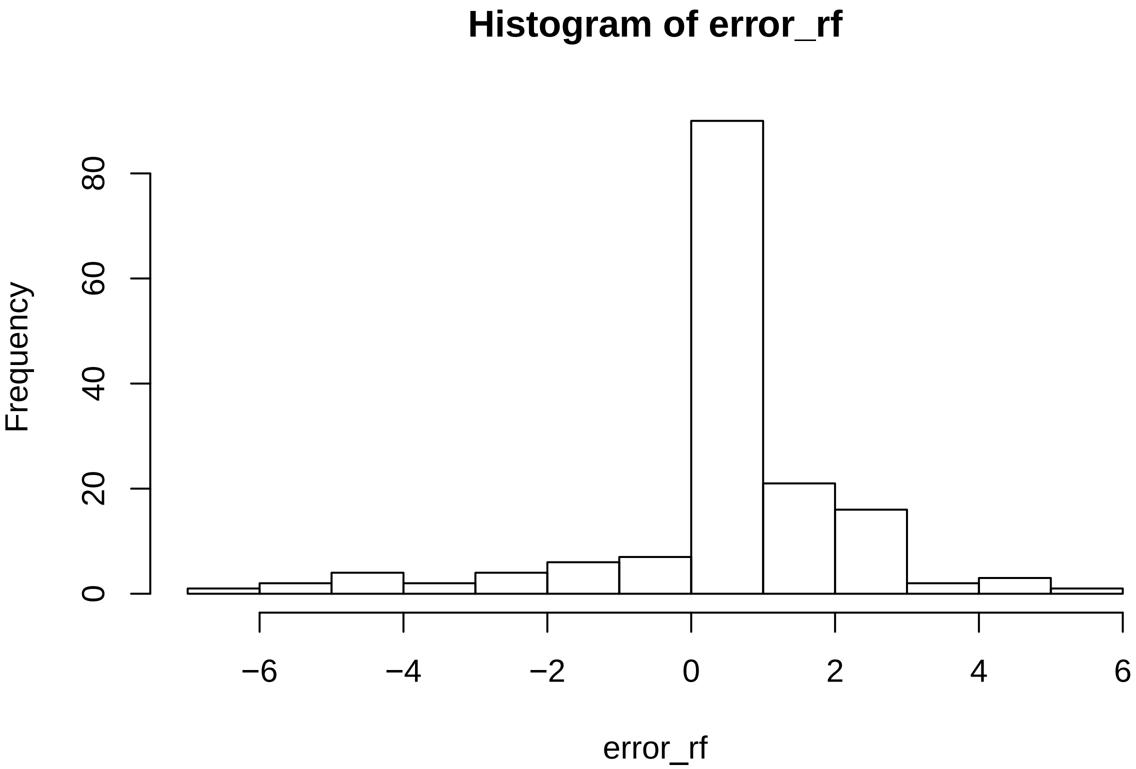
Elastic Net

1. Split data into train and test and use a for loop to find the best lambda value.
2. Create a new lm model with the coefficients selected in the best elastic net model
3. Find the cross validated mse

Random Forest

1. Create a random Forest model and get the mse

In random forests, there is no need for cross-validation or a separate test set to get an unbiased estimate of the test set error. It is estimated internally , during the run. source: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#ooberr



Results

	lm model	Random Forest
MSE	5.22e+00	3.26e+00

Random Forest had the smallest mse. I decided to use the split training and test data to create another model and obtain the error for the Random Forest on the training data. This error will be used as the distribution to select my values for perturbation. I did the same for the linear model just to compare the results. Results were relatively compatible, seen below.

```
## Warning in `[<-.data.table`(`*tmp*`, missing, 7, value = c(`1` = 7, `2`
## = 4, : Coerced 'double' RHS to 'integer' to match the factor column's
## underlying type. Character columns are now recommended (can be in keys), or
## coerce RHS to integer or character first.

## Warning in `[<-.data.table`(`*tmp*`, missing, 7, value = c(`1` = 7, `2`
## = 4, : Coerced 'double' RHS to 'integer' to match the factor column's
## underlying type. Character columns are now recommended (can be in keys), or
## coerce RHS to integer or character first.
```

```
##     RF LM RF_Perturbed LM_Perturbed
##  1:  7  4            7            5
##  2:  4  4            5            5
##  3:  2  2            3            3
##  4:  2  3            2            3
##  5:  3  2            3           -2
##  6:  2  3            2            3
##  7:  2  3            3           -2
##  8:  2  3            3            5
##  9:  2  3             6             4
## 10:  6  6             6             7
## 11:  2  2             2             3
## 12:  5  3             4             4
## 13:  5  4             8             5
## 14:  2  3             5             3
## 15:  2  2             2             5
## 16:  2  2             4             3
```

Campare the results and quality of classification models with the 5 data sets

I am going to try and use the caret package, which I have seen, but never used.

Methods

1. Split data into training and test
2. Use a cross-validated grid search to find best hyperparameter values (C for svm, K for knn)
3. Use test data to make a new prediction and confusion matrix

Results

	KNN	SVM
Mean	0.95	0.98
Mode	0.98	0.96
Regression	0.93	0.96
Perturb	0.96	0.96
Complete Case	0.96	0.99

Discussion

The best model and data set were the SVM complete cases combo. Although I expected the SVM model to outperform KNN I was surprised to see this. I will say that there were a lot of assumptions made about the data. Such as treating it as continuous and then rounding to the nearest integer. Also I followed a tutorial on the caret package and they split the data up into training and testing, ran cross-validation on the training and then used the testing set to choose a final accuracy. Although I enjoyed the tutorial and now really enjoy the caret package, I am not sure this demonstrates what the best model is. I think using the cross validation accuracy would be more appropriate. THe above results seem more based on luck of the draw.