STA 325L Final Report

Angela Wang, QiHan Zhou, Matthew Murray, Michelle Mao, Sara Lemus, Sophie Dalldorf

12/9/2021

Q.1 Prediction Question: Can we use linear regression to predict the change in a child's test scores that occur after watching Sesame street (or in some instances, not watching Sesame street)?

We decided to utilize three different types of models to gain inference and predict the difference in test scores that occurs after subjects watch Sesame Street: (1) least-squares regression model (2) ridge regression model and (3) regression tree model. There are six different test scores in the data set, so we fit three models for each difference in test score variable. Each model took the following variables as covariates:`site`, `sex`, `viewcat`, `setting`, `viewenc`. Before creating these models, we factored those variables to encode them as categoricals. The least-squares and ridge regression models also took into account all possible two-way interaction terms between the variables.

One problem that we envisioned when evaluating and comparing the different models is that the tests are scored on different scales. For example, the scores for the test on knowledge of body parts (noted by `bodyDiff`) range from 0-32, while those of the test on letters (noted by `letDiff`) range from 0-58. To be able to aptly compare the mean squared error (MSE) between models, we also decided to convert each response variable to the same range. More specifically, we scaled each variable to the arbitrary range [0, 30]. Lastly, we randomly split the data between testing and training, using 70% of the data for training and 30% of the data for testing.

We first decided to use least-squares regression models due to the fact that they provide apt inference into the relationship between the covariates and the response variable. Thereafter, we decided to use a ridge regression model due to the fact that ridge regression often provides performance improvements by shrinking slope coefficients. While shrinkage may introduce bias to a model, it decreases the variance and increases the precision of the slope coefficient estimates. The shrinkage is achieved by applying a shrinkage parameter, λ, to the Euclidean norm of a slope coefficient. Doing so slightly increases the bias of our model (as least-squares regression coefficient estimates are unbiased) but can also significantly decrease the variance (and increase the precision) of our regression coefficients. This slight increase in bias but significant decrease in variance usually decreases the MSE of a model.

We tuned our λ parameter using 10-fold cross validation. More specifically, we computed the cross-validation error rate for our model for a grid of λ values. Thereafter, we selected the λ value for which the cross-validation error is the smallest.

Initially, we were inclined to use least absolute shrinkage and selection operator (LASSO) regression. The main advantage of LASSO regression over ridge regression is that LASSO regression performs variable selection by setting the slope coefficients of inert predictors to 0. The reason why we initially thought that LASSO regression would work better than ridge regression is that in most of our linear models, only a small subset of variables are significant at a 95% significance level. However, our ridge regression models performed marginally better than our LASSO regression models, so for this reason, we decided to report the MSE's for the ridge regression models.

~Talk about why we chose regression tree models~

Table 1: Test Metrics

Response	Least.Regression.Test.MSE	Ridge.Regression.Test.MSE	Regression.Tree.Test.MSE
bodyDiff	32.45	21.61	20.60
letDiff	24.45	14.20	15.41
formDiff	23.19	12.83	14.92
numbDiff	28.24	14.63	15.91
relatDiff	23.64	19.86	19.89
clasfDiff	65.41	44.26	45.53

From the results of the above table, one can see that the ridge regression models have the best performance, although the performance of the regression trees are very similar . For each response variable except `bodyDiff`, the ridge regression model reports the lowest test MSE, although the difference in performance between the ridge regression and the regression tree is very marginal.

Q.2 Classification Question: Can we use the pre-test scores and other demographic variables to predict which region the children came from?

SVM

```
##   site  n
## 1    1 40
## 2    2 42
## 3    3 48
## 4    4 25
## 5    5 13
```

```
## Confusion Matrix and Statistics
##
##     pred
## true  1  2  3  4  5
##    1  2  5 13  0  0
##    2  0  8  5  0  0
##    3  1  1 14  0  0
##    4  0  4 14  0  0
##    5  0  1  4  0  0
##
## Overall Statistics
##
##                Accuracy : 0.3333
##                  95% CI : (0.2266, 0.4543)
##     No Information Rate : 0.6944
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1523
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
```

```
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity           0.66667   0.4211   0.2800       NA       NA
## Specificity           0.73913   0.9057   0.9091     0.75  0.93056
## Pos Pred Value        0.10000   0.6154   0.8750       NA       NA
## Neg Pred Value        0.98077   0.8136   0.3571       NA       NA
## Prevalence            0.04167   0.2639   0.6944     0.00  0.00000
## Detection Rate        0.02778   0.1111   0.1944     0.00  0.00000
## Detection Prevalence  0.27778   0.1806   0.2222     0.25  0.06944
## Balanced Accuracy     0.70290   0.6634   0.5945       NA       NA

## Confusion Matrix and Statistics
##
##     pred
## true  1  2  3  4  5
##    1  7  3 10  0  0
##    2  3  6  4  0  0
##    3  0  2 14  0  0
##    4  2  4 12  0  0
##    5  0  1  4  0  0
##
## Overall Statistics
##
##                Accuracy : 0.375
##                  95% CI : (0.2636, 0.497)
##     No Information Rate : 0.6111
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1964
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity           0.58333  0.37500   0.3182       NA       NA
## Specificity           0.78333  0.87500   0.9286     0.75  0.93056
## Pos Pred Value        0.35000  0.46154   0.8750       NA       NA
## Neg Pred Value        0.90385  0.83051   0.4643       NA       NA
## Prevalence            0.16667  0.22222   0.6111     0.00  0.00000
## Detection Rate        0.09722  0.08333   0.1944     0.00  0.00000
## Detection Prevalence  0.27778  0.18056   0.2222     0.25  0.06944
## Balanced Accuracy     0.68333  0.62500   0.6234       NA       NA

## Confusion Matrix and Statistics
##
##     pred
## true  1  2  3  4  5
##    1  1  7 12  0  0
##    2  1  8  4  0  0
##    3  3  1 12  0  0
##    4  3  4 11  0  0
##    5  0  1  4  0  0
##
## Overall Statistics
```

```
##
##                Accuracy : 0.2917
##                  95% CI : (0.1905, 0.4107)
##     No Information Rate : 0.5972
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0962
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity           0.12500   0.3810   0.2791       NA       NA
## Specificity           0.70312   0.9020   0.8621     0.75  0.93056
## Pos Pred Value        0.05000   0.6154   0.7500       NA       NA
## Neg Pred Value        0.86538   0.7797   0.4464       NA       NA
## Prevalence            0.11111   0.2917   0.5972     0.00  0.00000
## Detection Rate        0.01389   0.1111   0.1667     0.00  0.00000
## Detection Prevalence  0.27778   0.1806   0.2222     0.25  0.06944
## Balanced Accuracy     0.41406   0.6415   0.5706       NA       NA
```

```
## Confusion Matrix and Statistics
##
##     pred
## true  1  2  3  4  5
##    1  9  2  9  0  0
##    2  4  5  4  0  0
##    3  3  1 12  0  0
##    4  3  3 12  0  0
##    5  0  1  4  0  0
##
## Overall Statistics
##
##                Accuracy : 0.3611
##                  95% CI : (0.2512, 0.4829)
##     No Information Rate : 0.5694
##     P-Value [Acc > NIR] : 0.9999
##
##                   Kappa : 0.1703
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.4737  0.41667   0.2927       NA       NA
## Specificity            0.7925  0.86667   0.8710     0.75  0.93056
## Pos Pred Value         0.4500  0.38462   0.7500       NA       NA
## Neg Pred Value         0.8077  0.88136   0.4821       NA       NA
## Prevalence             0.2639  0.16667   0.5694     0.00  0.00000
## Detection Rate         0.1250  0.06944   0.1667     0.00  0.00000
## Detection Prevalence   0.2778  0.18056   0.2222     0.25  0.06944
## Balanced Accuracy      0.6331  0.64167   0.5818       NA       NA
```

```
## Confusion Matrix and Statistics
##
##     pred
## true  1  2  3  4  5
##    1  6  1  6  2  5
##    2  1  7  1  1  3
##    3  0  1 11  1  3
##    4  3  3  9  3  0
##    5  0  1  1  1  2
##
## Overall Statistics
##
##                Accuracy : 0.4028
##                  95% CI : (0.2888, 0.525)
##     No Information Rate : 0.3889
##     P-Value [Acc > NIR] : 0.44844
##
##                   Kappa : 0.2554
##
##  Mcnemar's Test P-Value : 0.01728
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity           0.60000  0.53846   0.3929  0.37500  0.15385
## Specificity           0.77419  0.89831   0.8864  0.76562  0.94915
## Pos Pred Value        0.30000  0.53846   0.6875  0.16667  0.40000
## Neg Pred Value        0.92308  0.89831   0.6964  0.90741  0.83582
## Prevalence            0.13889  0.18056   0.3889  0.11111  0.18056
## Detection Rate        0.08333  0.09722   0.1528  0.04167  0.02778
## Detection Prevalence  0.27778  0.18056   0.2222  0.25000  0.06944
## Balanced Accuracy     0.68710  0.71838   0.6396  0.57031  0.55150

## Confusion Matrix and Statistics
##
##     pred
## true  1  2  3  4  5
##    1  6  2  7  4  1
##    2  4  4  3  0  2
##    3  1  1 11  3  0
##    4  2  2  9  4  1
##    5  0  2  1  1  1
##
## Overall Statistics
##
##                Accuracy : 0.3611
##                  95% CI : (0.2512, 0.4829)
##     No Information Rate : 0.4306
##     P-Value [Acc > NIR] : 0.9056
##
##                   Kappa : 0.181
##
##  Mcnemar's Test P-Value : 0.1807
##
```

```
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity           0.46154  0.36364   0.3548  0.33333  0.20000
## Specificity           0.76271  0.85246   0.8780  0.76667  0.94030
## Pos Pred Value        0.30000  0.30769   0.6875  0.22222  0.20000
## Neg Pred Value        0.86538  0.88136   0.6429  0.85185  0.94030
## Prevalence            0.18056  0.15278   0.4306  0.16667  0.06944
## Detection Rate        0.08333  0.05556   0.1528  0.05556  0.01389
## Detection Prevalence  0.27778  0.18056   0.2222  0.25000  0.06944
## Balanced Accuracy     0.61213  0.60805   0.6164  0.55000  0.57015

## Confusion Matrix and Statistics
##
##     pred
## true  1  2  3  4  5
##    1  1  1  6  9  3
##    2  1  3  4  5  0
##    3  1  0  4 11  0
##    4  0  0  7 11  0
##    5  0  0  3  2  0
##
## Overall Statistics
##
##                Accuracy : 0.2639
##                  95% CI : (0.167, 0.381)
##     No Information Rate : 0.5278
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0434
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity           0.33333  0.75000  0.16667   0.2895  0.00000
## Specificity           0.72464  0.85294  0.75000   0.7941  0.92754
## Pos Pred Value        0.05000  0.23077  0.25000   0.6111  0.00000
## Neg Pred Value        0.96154  0.98305  0.64286   0.5000  0.95522
## Prevalence            0.04167  0.05556  0.33333   0.5278  0.04167
## Detection Rate        0.01389  0.04167  0.05556   0.1528  0.00000
## Detection Prevalence  0.27778  0.18056  0.22222   0.2500  0.06944
## Balanced Accuracy     0.52899  0.80147  0.45833   0.5418  0.46377

## Confusion Matrix and Statistics
##
##     pred
## true  1  2  3  4  5
##    1  6  2  5  1  6
##    2  4  2  3  0  4
##    3  1  0 11  1  3
##    4  3  2  8  1  4
##    5  1  1  2  0  1
```

```
##
## Overall Statistics
##
##                Accuracy : 0.2917
##                  95% CI : (0.1905, 0.4107)
##     No Information Rate : 0.4028
##     P-Value [Acc > NIR] : 0.981095
##
##                   Kappa : 0.1226
##
##  Mcnemar's Test P-Value : 0.006726
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity           0.40000  0.28571   0.3793  0.33333  0.05556
## Specificity           0.75439  0.83077   0.8837  0.75362  0.92593
## Pos Pred Value        0.30000  0.15385   0.6875  0.05556  0.20000
## Neg Pred Value        0.82692  0.91525   0.6786  0.96296  0.74627
## Prevalence            0.20833  0.09722   0.4028  0.04167  0.25000
## Detection Rate        0.08333  0.02778   0.1528  0.01389  0.01389
## Detection Prevalence  0.27778  0.18056   0.2222  0.25000  0.06944
## Balanced Accuracy     0.57719  0.55824   0.6315  0.54348  0.49074
```

In our initial SVM models, Radial Kernel SVM came with the best performance in terms of prediction accuracy, but it was only slightly above 0.30. We came across several online resources that stated standardizing the variables and encoding the categorical variables have empirically imrpoved SVM models performance (https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf). After we standardized the variables and encoded the sex variable (if a data is a male, then it has (1,0)), we observed a light improvement in prediction accuracy across all models. However, one problem that particularly piqued our interests is that no predictions on class 4 class 5 are made before & after variable transformation. We then realized that, relatively, class 4 & 5 have less number of data than the other classes. Although we'd argue that the proportion not severe enough for us to deem the dataset as an imbalanced one, we used the formula below to assign weights to each class and specify "one versus one" comparison, which has been suggested to yield better prediction than "one versus all."

$w_j = \frac{n}{kn_j}$, n is total number of data points, k is number of classes, n_j is the number of data in class j

After class weight assignment, the SVM models began to make prediction on class 4 & 5, at the cost of overall accuracy that more test data are being misclassified even though a few number of 4&5 are correctly classified. Then, we began to experiment with the class weights and increase the class 4 & 5 weights by roughly 0.5, which is arbitrarily chosen, and it boosted linear kernel SVM's prediction accuracy to 0.403.