STAT479 Final Project: Speaker Accent Recognition

Fangying Zhan

December 18, 2020

1. Introduction

Different accents in the same language are distinguishable to the human ear, as they are to machines. The main goal of this project is to design and train models to recognize human accents. Unlike the human ear, which receives sound signals as input, statistical models usually requires numerical or categorical data. Consequently, raw data of voices need to be transformed into numerical data for models and algorithms. Mel-Frequency Cepstral Coefficient (MFCC) is one of the algorithms for voice signal feature extraction. Considering humans' hearing ability, it transforms and maps the signal in hertz onto Mel-scale with a threshold. By using feature extraction via MFCC, raw signals stored in audio files can be converted into numerical data, where patterns exist and can be recognized by models and algorithms.

Following a Speaker Accent Recognition research conducted by Ma and Fokou (2015), we will build models on the *Speaker Accent Recognition* dataset. Following their suggested approach, it might be useful to not only extract the means of MFCCs, but also take standard deviations into account in predicting the US and non-US accents. Since we will also be interested in the generalization performance of the best model, the approaches will be examined when we attempt on a new dataset - *Speech Accent Archive*, which contains raw sound data that might require further transformation, exploration and assessment.

Dataset Descriptions

UCI Dataset The first dataset is the *Speaker Accent Recognition* dataset from UCI Machine Learning Repository. The dataset contains 329 speech samples and consists of 12 explanatory variables that are obtained using MFCC on the original time domain soundtrack of reading a random word, and a response variable of 6 language accent labels.

Kaggle Dataset The second dataset is the *Speech Accent Archive* data from Kaggle, which contains 2140 speech samples of different talkers reading the same passage. The raw data still requires further processing steps to match the format of explanatory variables as in the UCI dataset. A response variable is already clear, as all names of 214 different native languages of the speakers are known. Besides, demographic information such as age, gender and birthplace are also recorded.

According to the relevant research conducted by Ma and Fokou (2015), in which the UCI dataset is mainly involved, the explanatory variables are mean values of 12 MFCCs. They also claimed that in practice, the first 13 MFCCs are usually computed. However, the reason why 12 instead of 13 MFCCs are included in this dataset is unclear. Since the UCI dataset also includes 10 samples of soundtrack for reference, it is possible for us to check that the 12 MFCCs included in the dataset are actually MFCCs from cepstrum #2 to #13, i.e., the #1 MFCC is excluded in this dataset. Even though the specific reason for choosing #2 to #13 MFCCs remains unknown, we can still explore the implications of this dataset, and also explore the Kaggle dataset following the same or a similar approach.

Data Preprocessing & Exploratory Data Analysis (EDA)

Feature Extraction As discussed above, feature extraction via MFCCs is needed for preprocessing the Kaggle dataset. Means and standard deviations are computed for each of MFCC spectra (#1-#13). For the

Table 1: Counts in datasets

	non-US	US
table.uci	164	165
table.kag	1703	390
table.fil	180	390

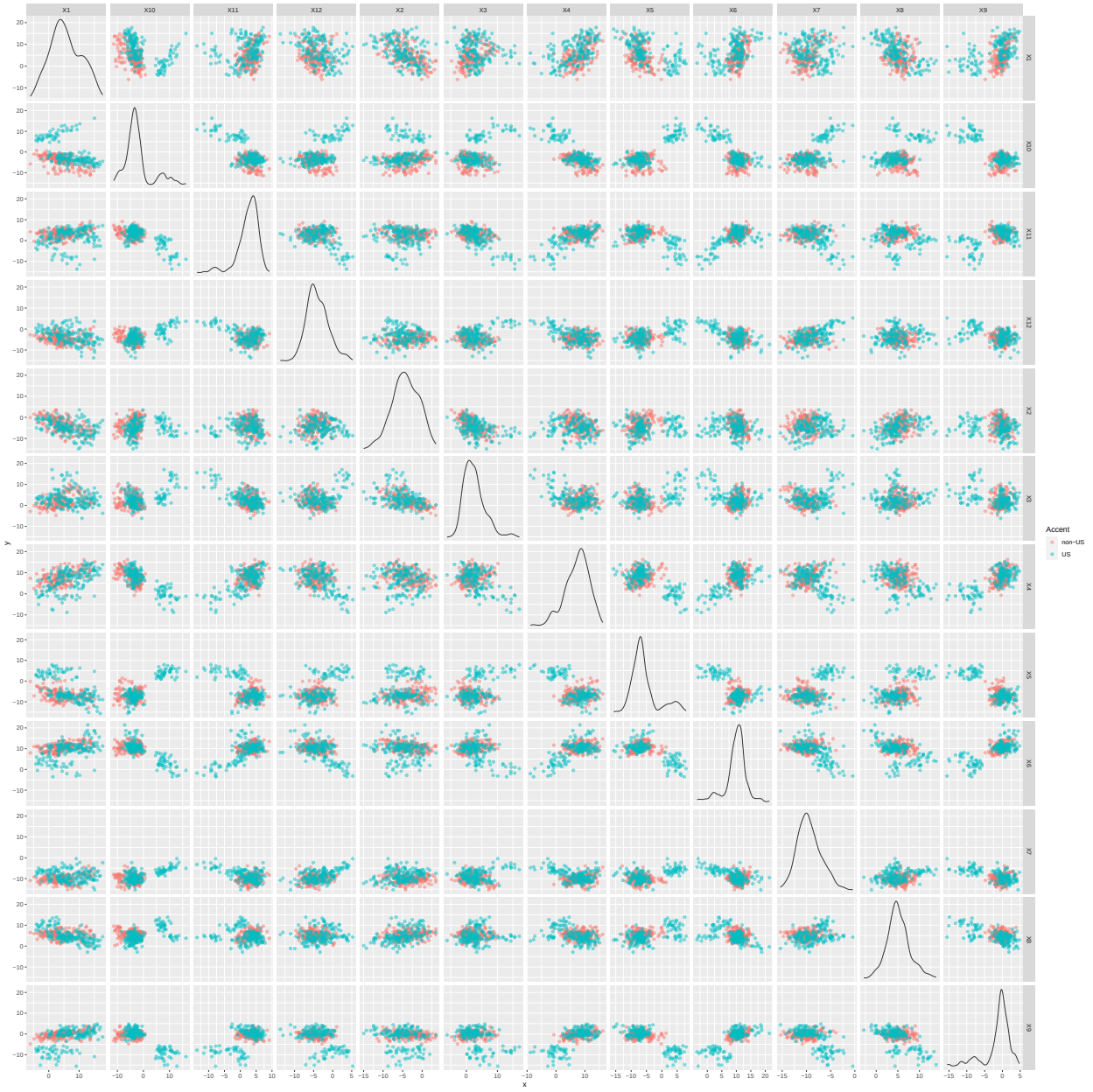
Kaggle dataset, we label them as feature variables $X1.mean, ..., X13.mean, X1.sd, ..., X13.sd$, whereas for the filtered version, we label means of #2-#13 as $X1, ..., X12$ to match the format of the UCI dataset.

Encoding Response y Since there are 6 accent class labels in the UCI dataset, which only has a small sample size of 329, and there are 214 accent class labels in the Kaggle dataset, which has a sample size of 2140, it is more convenient if we start from considering fewer classes. In this project, we would only perform binary classification, classifying talkers into US or non-US accent.

For the response variable y, we may naturally denote US accent as $y = 1$, and non-US accent as $y = 0$. In the UCI dataset, $y = 0$ is equivalent to the original data labels of the set {ES, FR, GE, IT, UK}, and $y = 1$ is equivalent to the label {US}. In the Kaggle dataset, however, only 1 label out of 214 native language labels is equivalent to $y = 1$, that is label {usa}. This would result in highly imbalanced class labels. One possible solution is to follow the UCI dataset format to filter out all class labels other than the 6 accent classes, and thus we may have a filtered version of dataset. The counts of class labels in the three datasets are as follows.

Encoding Feature X For the feature variables $X1, ..., X12$ in UCI dataset, which are numerical values extracted via MFCCs, distributions and pairwise scatter plots are shown as below. We may also plot for the Kaggle dataset. Other plots such as box plots and density plots for individual features are included in Appendix 2. Based on the exploratory analysis, it is clear that most of the variables are highly skewed, which is further confirmed via computing the skewness value in R. Thus, log or cubic transformation might be helpful for the data. Since data points tend to cluster, non-parametric methods might also be helpful.

Data Transformation Skewness values can be computed for each feature variables. For highly skewed variables, log, cube root, and square transformations are performed correspondingly. Models fitted on transformed version of data will be compared with models fitted on the original version, so that we may see if data transformation steps are useful in this context.



2. Methods

Models for classification

To solve this classification task, two methods are proposed, along with model selection techniques.

Model I: Generalized Linear Model (GLM) & Lasso For 12 continuous features and a binary response, it is convenient to make use of GLM by specifying the binomial family. A baseline model is to include only main effects in GLM. Besides, main effects, pairwise interactions will also be attempted if it is possible to perform. To perform variable selection, Lasso regularization is a convenient choice.

Model II: KNN Based on the previous analysis as well as the approach suggested by Ma and Fokou (2015), non-parametric methods should also be considered in this case. A convenient choice is the k Nearest

Neighbors model. For hyperparameter tuning, cross-validation is a popular method and thus will be used for tuning k.

Dataset Splitting

To select from all the models described in above and to evaluate the final model, dataset needs to be split into a Train set, a Test set, and a validation set. For this project, since 8:2 is a common ratio for splitting the dataset, we first split 80% as (Train + Test) and 20% as validation set, and then split the 80% fold into 80% Train set and 20% Test set. Specifically, to maintain the distributions in each dataset, stratified splitting is employed.

3. Results

Models on UCI dataset

GLM GLM is trained both on the original UCI training set data and the transformed version of data. In both cases, with only main effects included, only features $X1$ and $X10$ are statistically significant. The results are as follows:

	Training Acc.	Testing Acc.
GLM (MEs)	77.36%	78.85%
GLM (MEs + transformed)	76.42%	71.15%

Thus, GLM fitted on the original data set is preferred, with higher accuracy.

An attempt is to include pairwise interaction terms into GLM. However, it is not available as the algorithm failed to converge. Thus, we continue on Lasso with the main effects.

By performing Lasso regularization with 20-fold cross validation, an alpha value of 0.025 is selected. Standardization is also an option that is considered, but it does not give better results in the previous case. The best Lasso model gives the results as follows:

GLM	Training Acc.	Testing Acc.
Lasso (MEs)	75.00%	75.00%
Lasso (MEs + standardized)	69.34%	63.46%

Thus, Lasso method does not give a better performance than the previous GLM (MEs) model.

KNN By performing 20-fold cross-validation, the best KNN selected is k=5. Since preprocessing techniques including centering and scaling are important for KNN, in this case, they are also performed. The model results are as follows:

KNN	Training Acc.	Testing Acc.
5-Nearest Neighbor	83.96%	84.62%

Evaluation on Validation Set

Comparing the accuracy from models in above, KNN with k=5 is selected as the best model. To validate the accuracy, the model is evaluated on the validation set. The result is as follows:

Final Model	Validation Acc.	95% CI
5-Nearest Neighbor	83.08%	(0.7173, 0.9124)

Generalization Performance

We are always interested in generalization performance of our trained model. In this case, since we also have another useful dataset which is from a distinct source and is originally raw data, it would be interesting if we can see the performance of the KNN model trained on full UCI dataset on some unseen data. The result is as follows:

Kaggle dataset	Prediction Acc.	95% CI
UCI 5-Nearest Neighbor	62.98%	(0.5887, 0.6696)

It is clear that comparing to the previous validation accuracy of 83.08%, the prediction accuracy on Kaggle dataset is significantly low, which is only 62.98%. Following the same approach, we can also train and select models on the Kaggle dataset for further exploration.

Models on Kaggle Dataset

Predicting Accent

Using the same approach as above, we can also train different models and select from them. In this case, we use the filtered Kaggle dataset, which has 570 observations after cleaning and transforming the data. Again, we are interested in predicting the accent of speakers. Following Ma and Fokou's suggestions on future work (2015), this time feature variables include not only the means of MFCCs #1-#13, but also standard deviations. Consequently, there are 26 features in total. The results of models are as follows:

	Training Acc.	Testing Acc.
GLM (MEs)	73.91%	70.45%
GLM (MEs + transformed)	74.46%	70.45%
Lasso (MEs)	74.46%	71.59%
Lasso (MEs + standardized)	73.37%	70.45%
KNN (k=11)	70.38%	68.18%

Again, in this case, GLM failed to include pairwise interaction terms since the algorithm cannot converge. Selecting from the model performance above, Lasso models with main effects gives the highest testing accuracy. The result of evaluating Lasso model on validation set is as follows:

Final Model	Validation Acc.	95% CI
Lasso (MEs)	61.40%	(0.5183, 0.7037)

Compared with models on the UCI dataset, the models on Kaggle dataset seems to give worse performance. It could be the fact that Kaggle dataset is extracted from raw data, which contains much more noises than UCI dataset. Without any expertise in MFCCs and feature extractions, there is no way to find out the specific reasons for such bad model performance on noisy data. Yet, we may further explore the Kaggle dataset in predicting the gender of the talkers. Unlike predicting accent, predicting gender could be an easier task so that we may find out whether the Kaggle dataset is useful.

Table 9: Counts in datasets

	female	male
table.gen	1014	1079

Predicting Gender

Using the same approach, we can make use of the full Kaggle dataset with 2093 observations, split the data, train models and perform selections, and finally evaluate the best model. The binary gender information is also provided by the Kaggle dataset, so in this case, we may perform binary classification.

The 26 feature variables are, again, means and standard deviations of MFCCs #1-#13. The model results are as follows:

	Training Acc.	Testing Acc.
GLM (MEs)	83.33%	83.13%
GLM (MEs + transformed)	84.30%	83.13%
Lasso (MEs)	82.81%	80.72%
Lasso (MEs + standardized)	82.37%	81.02%
KNN (k=13)	77.83%	76.81%

The best model for predicting binary gender is either GLM on the original data or GLM on transformed data. Here we choose GLM with main effects on the original data, since it suffers less from overfitting. The result of evaluating GLM on validation set is as follows:

Final Model	Validation Acc.	95% CI
GLM (MEs)	83.21%	(0.7927, 0.8667)

Compared with previous model performance in accent predicting tasks, the model for predicting binary gender gives consistent performance and the prediction accuracy is comparatively high.

4. Discussion

In this project, two types of models are mainly used in the task of predicting accent of speakers given the MFCCs mean and standard deviation data: Generalized Linear Model and K-Nearest Neighbors. Lasso regularization is employed together with GLM for model selection. K-Fold Cross Validation is utilized in hyperparameter tuning. Based on the model performances, KNN performs well on predicting accent of speakers in UCI dataset, which align with the exploratory analysis where data points appear to be non-linear, so that non-parametric approaches might be useful. Lasso models do not give better performances, indicating that all features are useful in predicting speaker accent. By validating the model on an independent set of UCI data, we can confirm that the final model gives considerably stable performance.

When attempting to evaluate the generalization performance of the best model on a different set of data - the Kaggle dataset, however, we see that the prediction accuracy significantly falls, which implies that it is not enough to only train models on a small dataset of 329 observations in solving the problem of accent recognition. Besides the issue of dataset size, another question that remains unsolved is that data from soundtracks can be very noisy, resulting in trained models that fail to capture the patterns. The Kaggle dataset is such an example where the task of predicting accent is much more difficult to solve. Yet, such noisy dataset is still useful in solving other tasks that might be much easier, such as predicting gender.

Based on our exploration, feature extraction appears to be a very important step in solving the problem of speaker accent recognition. Future directions should be aiming at not only making use of the means and standard deviations of MFCCs, but also extracting patterns of MFCCs. Data including only means and standard deviations are not enough in training models that can give better performance.

Appendix 1: Model Outputs

Part 1: explore UCI Dataset

UCI Dataset: predict accent

Data transformation

```
## [1] -0.2531251
```

```
## [1] -0.6255283
```

```
## [1] 0.3571482
```

```
## [1] 0.07530183
```

```
## [1] 0.08796726
```

```
## [1] -0.4882947
```

splitting data: UCI dataset

Model training

```
##
## Call:
## glm(formula = y.train ~ (X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 +
##     X9 + X10 + X11 + X12), family = "binomial")
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.13685  -0.70050  -0.05315   0.62830   2.21230
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  6.62990    3.73743   1.774   0.0761 .
## X1           0.32749    0.07618   4.299 1.72e-05 ***
## X2          -0.00398    0.09058  -0.044   0.9650
## X3           0.18537    0.11706   1.584   0.1133
## X4          -0.12236    0.11432  -1.070   0.2845
## X5           0.20957    0.12809   1.636   0.1018
## X6          -0.25016    0.13309  -1.880   0.0602 .
## X7           0.10671    0.13511   0.790   0.4297
## X8          -0.24453    0.14310  -1.709   0.0875 .
## X9           0.11329    0.14055   0.806   0.4202
## X10          0.43038    0.11035   3.900 9.62e-05 ***
## X11          0.14265    0.12233   1.166   0.2436
## X12          0.13955    0.08347   1.672   0.0946 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 293.89  on 211  degrees of freedom
## Residual deviance: 182.30  on 199  degrees of freedom
## AIC: 208.3
##
## Number of Fisher Scoring iterations: 8
```

```
##
## Call:
## glm(formula = y.train ~ (X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 +
##     X9 + X10 + X11 + X12), family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8537  -0.7119  -0.0502   0.5912   2.4086
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.79570    3.03150   0.592   0.5536
## X1           0.33337    0.08505   3.919 8.87e-05 ***
## X2          -0.06642    0.07451  -0.891   0.3727
## X3           0.94359    0.87576   1.077   0.2813
## X4          -0.02267    0.10029  -0.226   0.8211
## X5          -0.26408    0.33136  -0.797   0.4255
## X6          -0.12835    0.11102  -1.156   0.2476
## X7          -0.01053    0.11879  -0.089   0.9294
## X8          -0.11881    0.11981  -0.992   0.3214
## X9          -0.15740    0.22168  -0.710   0.4777
## X10          0.50838    0.10941   4.647 3.38e-06 ***
## X11          0.13530    0.27219   0.497   0.6191
## X12          0.17198    0.08892   1.934   0.0531 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 293.89  on 211  degrees of freedom
## Residual deviance: 183.75  on 199  degrees of freedom
## AIC: 209.75
##
## Number of Fisher Scoring iterations: 7
```

Training set misclassification error

```
## UCI original data:
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 87 19
##          1 29 77
##
##                Accuracy : 0.7736
##                  95% CI : (0.7113, 0.8281)
##     No Information Rate : 0.5472
##     P-Value [Acc > NIR] : 6.113e-12
##
##                   Kappa : 0.5472
##
##  Mcnemar's Test P-Value : 0.1939
##
```

```
##             Sensitivity : 0.7500
##             Specificity : 0.8021
##          Pos Pred Value : 0.8208
##          Neg Pred Value : 0.7264
##              Prevalence : 0.5472
##          Detection Rate : 0.4104
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.7760
##
##        'Positive' Class : 0
##
```

```
## UCI transformed data:
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 85 21
##          1 29 77
##
##                Accuracy : 0.7642
##                  95% CI : (0.7012, 0.8196)
##     No Information Rate : 0.5377
##     P-Value [Acc > NIR] : 7.716e-12
##
##                   Kappa : 0.5283
##
##  Mcnemar's Test P-Value : 0.3222
##
##             Sensitivity : 0.7456
##             Specificity : 0.7857
##          Pos Pred Value : 0.8019
##          Neg Pred Value : 0.7264
##              Prevalence : 0.5377
##          Detection Rate : 0.4009
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.7657
##
##        'Positive' Class : 0
##
```

Performance on test set

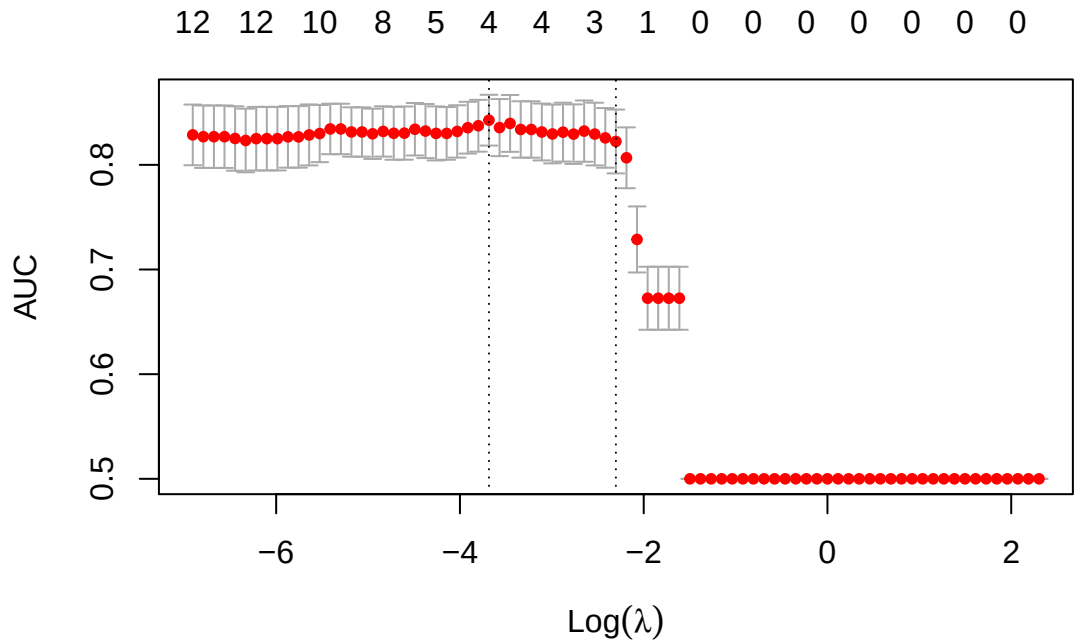
```
## UCI original data:
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 21  5
##          1  6 20
##
##                Accuracy : 0.7885
##                  95% CI : (0.653, 0.8894)
##     No Information Rate : 0.5192
```

```
##     P-Value [Acc > NIR] : 5.622e-05
##
##                   Kappa : 0.5769
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.7778
##             Specificity : 0.8000
##          Pos Pred Value : 0.8077
##          Neg Pred Value : 0.7692
##              Prevalence : 0.5192
##          Detection Rate : 0.4038
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.7889
##
##        'Positive' Class : 0
##
```

```
## UCI transformed data:
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 20  6
##          1  9 17
##
##                Accuracy : 0.7115
##                  95% CI : (0.5692, 0.8287)
##     No Information Rate : 0.5577
##     P-Value [Acc > NIR] : 0.01677
##
##                   Kappa : 0.4231
##
##  Mcnemar's Test P-Value : 0.60558
##
##             Sensitivity : 0.6897
##             Specificity : 0.7391
##          Pos Pred Value : 0.7692
##          Neg Pred Value : 0.6538
##              Prevalence : 0.5577
##          Detection Rate : 0.3846
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.7144
##
##        'Positive' Class : 0
##
```



LASSO selection

```
## [1] 0.02511886
```

LASSO best lambda

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 97  9
##          1 44 62
##
##                Accuracy : 0.75
##                  95% CI : (0.6861, 0.8068)
##     No Information Rate : 0.6651
##     P-Value [Acc > NIR] : 0.004666
##
##                   Kappa : 0.5
##
##  Mcnemar's Test P-Value : 3.008e-06
##
##             Sensitivity : 0.6879
##             Specificity : 0.8732
##          Pos Pred Value : 0.9151
##          Neg Pred Value : 0.5849
##              Prevalence : 0.6651
##          Detection Rate : 0.4575
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.7806
##
```

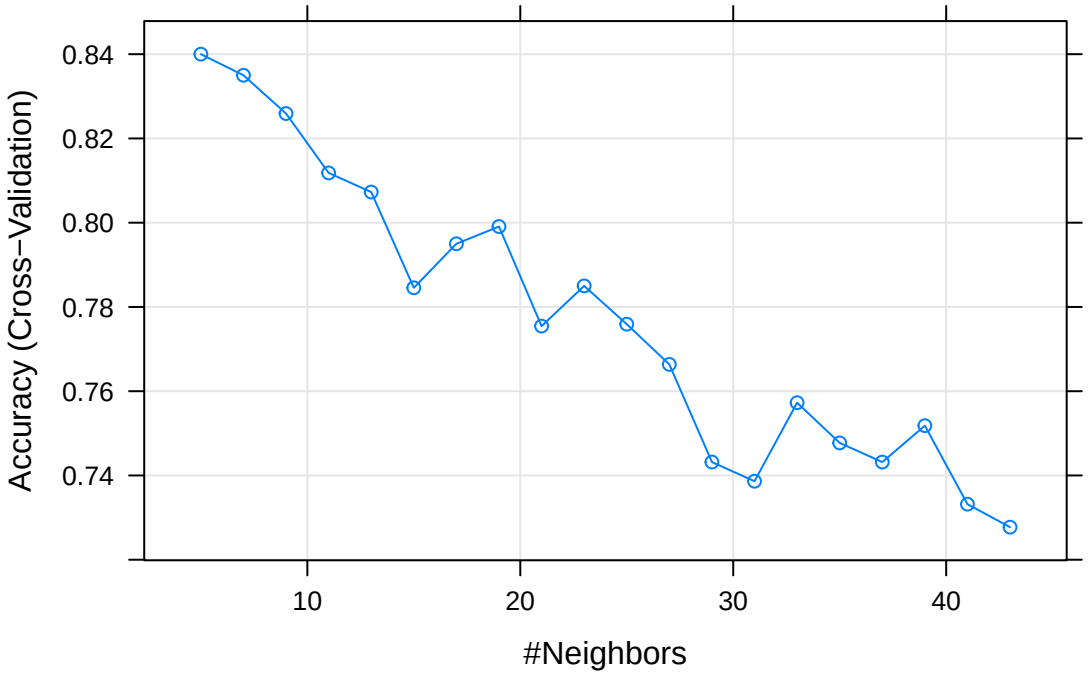
```
##        'Positive' Class : 0
##
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 23  3
##          1 10 16
##
##                Accuracy : 0.75
##                  95% CI : (0.6105, 0.8597)
##     No Information Rate : 0.6346
##     P-Value [Acc > NIR] : 0.05377
##
##                   Kappa : 0.5
##
##  Mcnemar's Test P-Value : 0.09609
##
##             Sensitivity : 0.6970
##             Specificity : 0.8421
##          Pos Pred Value : 0.8846
##          Neg Pred Value : 0.6154
##              Prevalence : 0.6346
##          Detection Rate : 0.4423
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.7695
##
##        'Positive' Class : 0
##
```

KNN with K-Fold CV

```
## k-Nearest Neighbors
##
## 212 samples
##  12 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (12), scaled (12)
## Resampling: Cross-Validated (20 fold)
## Summary of sample sizes: 201, 201, 201, 201, 201, 202, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.8400000  0.6806882
##    7  0.8350000  0.6715705
##    9  0.8259091  0.6532141
##   11  0.8118182  0.6245982
##   13  0.8072727  0.6153062
##   15  0.7845455  0.5724576
##   17  0.7950000  0.5920936
##   19  0.7990909  0.6016930
##   21  0.7754545  0.5549347
##   23  0.7850000  0.5731212
```

```
##    25  0.7759091  0.5537506
##    27  0.7663636  0.5338224
##    29  0.7431818  0.4867572
##    31  0.7386364  0.4771244
##    33  0.7572727  0.5172402
##    35  0.7477273  0.4966013
##    37  0.7431818  0.4896804
##    39  0.7518182  0.5055121
##    41  0.7331818  0.4684528
##    43  0.7277273  0.4588126
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```




```
## 5-nearest neighbor model
## Training set outcome distribution:
##
##   0   1
## 106 106
```

```
## Cross-Validated (20 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    0    1
##          0 43.9  9.9
##          1  6.1 40.1
##
##  Accuracy (average) : 0.8396
```

KNN on testing set

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 24  6
##          1  2 20
##
##                Accuracy : 0.8462
##                  95% CI : (0.7192, 0.9312)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 2.02e-07
##
##                   Kappa : 0.6923
##
##  Mcnemar's Test P-Value : 0.2888
##
##             Sensitivity : 0.9231
##             Specificity : 0.7692
##          Pos Pred Value : 0.8000
##          Neg Pred Value : 0.9091
##              Prevalence : 0.5000
##          Detection Rate : 0.4615
##    Detection Prevalence : 0.5769
##       Balanced Accuracy : 0.8462
##
##        'Positive' Class : 0
##
```

validation on best model

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 29  8
##          1  3 25
##
##                Accuracy : 0.8308
##                  95% CI : (0.7173, 0.9124)
##     No Information Rate : 0.5077
##     P-Value [Acc > NIR] : 5.838e-08
##
##                   Kappa : 0.6623
##
##  Mcnemar's Test P-Value : 0.2278
##
##             Sensitivity : 0.9062
##             Specificity : 0.7576
##          Pos Pred Value : 0.7838
##          Neg Pred Value : 0.8929
##              Prevalence : 0.4923
##          Detection Rate : 0.4462
##    Detection Prevalence : 0.5692
```

```
##       Balanced Accuracy : 0.8319
##
##        'Positive' Class : 0
##
```

Final model

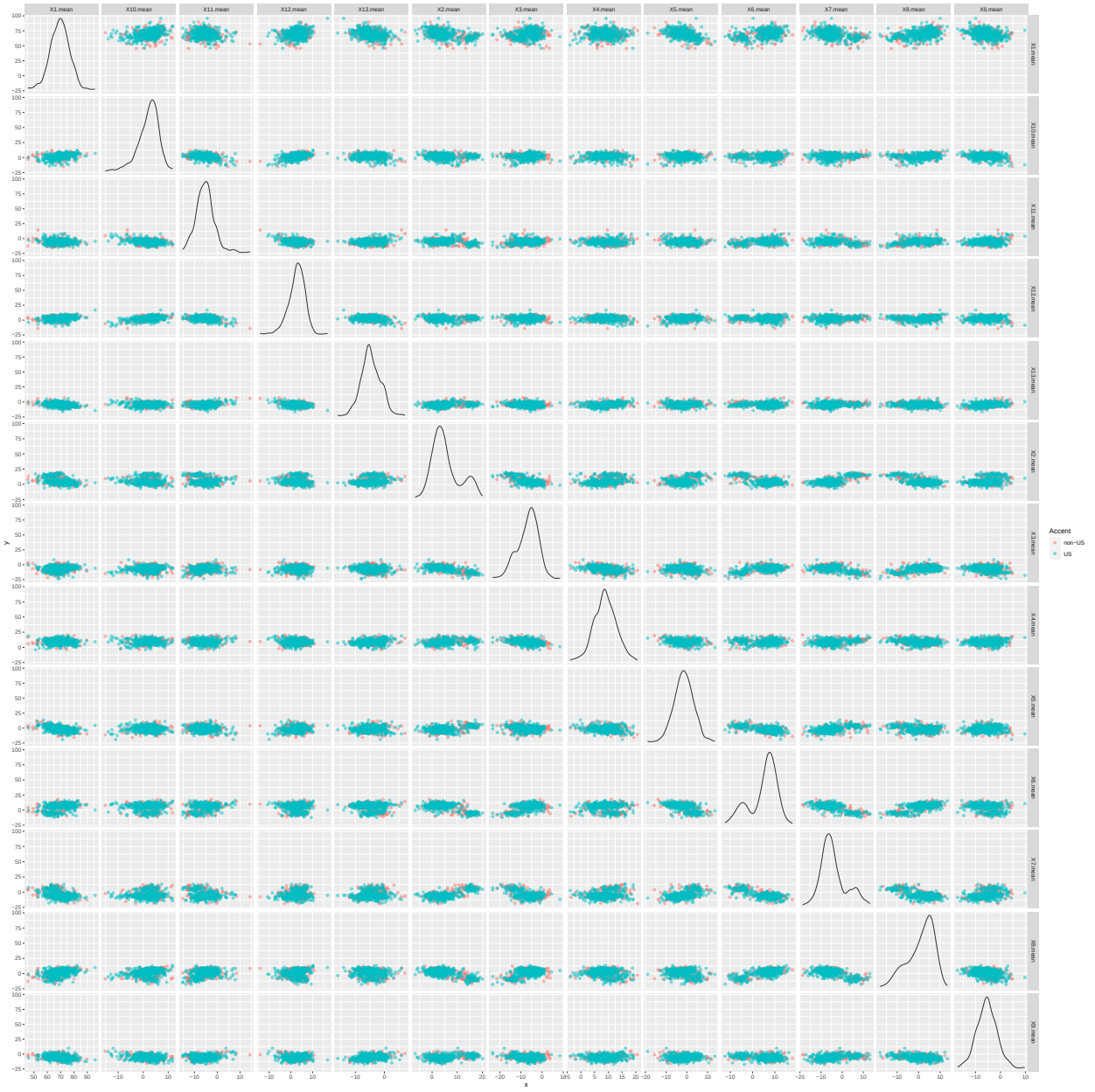
```
## k-Nearest Neighbors
##
## 329 samples
##  12 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (12), scaled (12)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 297, 296, 296, 295, 295, 296, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.8418171  0.6835707
##    7  0.8305128  0.6610076
##    9  0.8227328  0.6453473
##   11  0.8217525  0.6434086
##   13  0.8197322  0.6395330
##   15  0.8165794  0.6329653
##   17  0.8065972  0.6129886
##   19  0.7995822  0.5989903
##   21  0.7933025  0.5865878
##   23  0.7955065  0.5910741
##   25  0.7913417  0.5826150
##   27  0.7759637  0.5518915
##   29  0.7759655  0.5519358
##   31  0.7747400  0.5495371
##   33  0.7790590  0.5582909
##   35  0.7748013  0.5498280
##   37  0.7779226  0.5560855
##   39  0.7697155  0.5397878
##   41  0.7707851  0.5419813
##   43  0.7716726  0.5436731
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

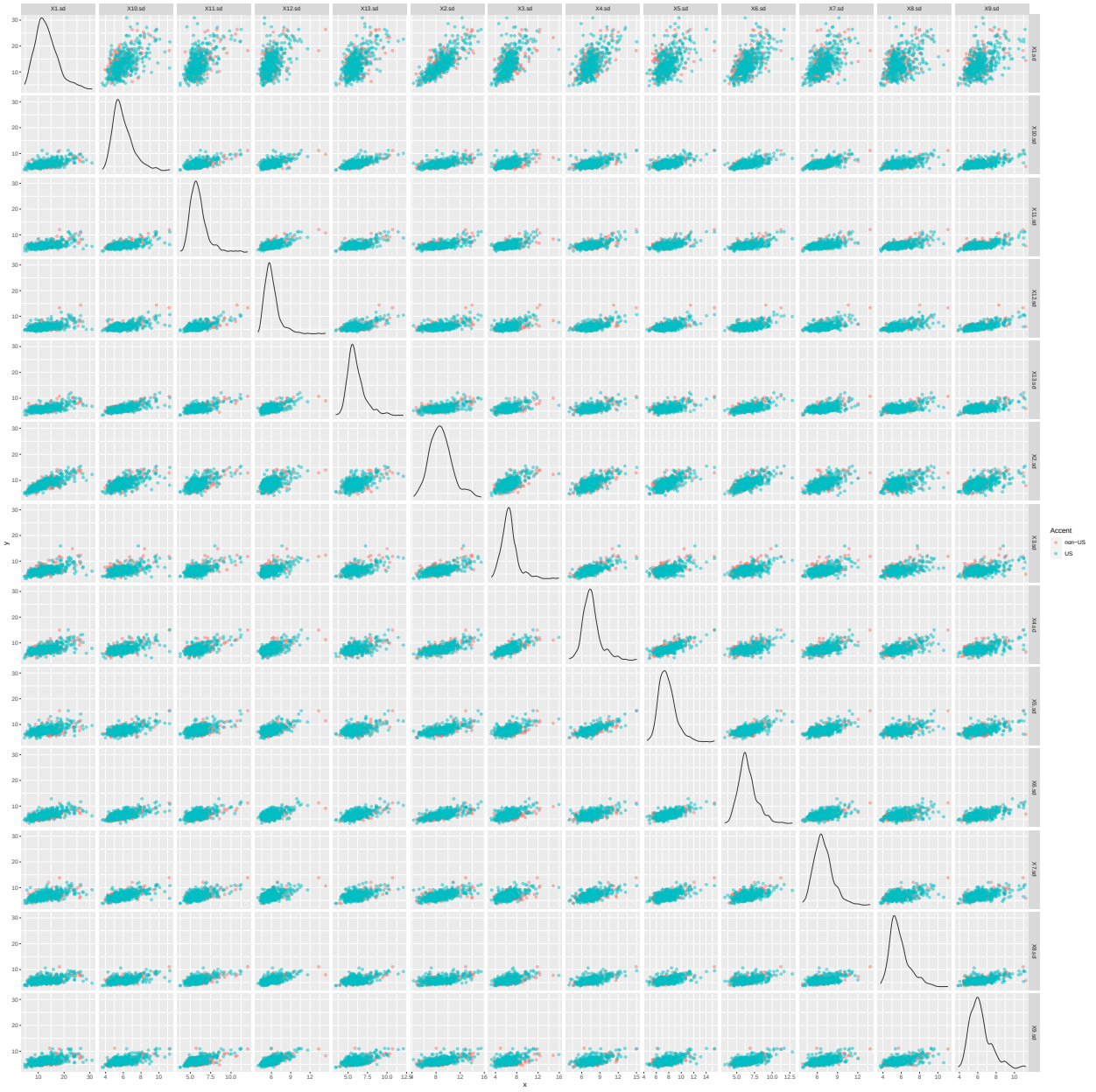
Final model: generalization performance on Kaggle data

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  11  42
##          1 169 348
##
##                Accuracy : 0.6298
##                  95% CI : (0.5887, 0.6696)
##     No Information Rate : 0.6842
```

```
##     P-Value [Acc > NIR] : 0.9975
##
##                   Kappa : -0.0575
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.06111
##             Specificity : 0.89231
##          Pos Pred Value : 0.20755
##          Neg Pred Value : 0.67311
##              Prevalence : 0.31579
##          Detection Rate : 0.01930
##    Detection Prevalence : 0.09298
##       Balanced Accuracy : 0.47671
##
##        'Positive' Class : 0
##
```

Part 2: explore Kaggle dataset





Kaggle dataset: predict accent

Data transformation

```
## [1] -0.1020955
## [1] 0.2478505
## [1] 0.3053799
## [1] 0.4586376
## [1] 0.2114597
## [1] 0.1344023
## [1] 0.6426798
```

```
## [1] 0.60529
```

```
## [1] 0.6413364
```

```
## [1] 0.9156241
```

```
## [1] 0.7250354
```

```
## [1] 0.7072888
```

Split Kaggle dataset: training, testing & validation sets

glm on training data

```
##
## Call:
## glm(formula = y.train ~ X1.mean + X2.mean + X3.mean + X4.mean +
##     X5.mean + X6.mean + X7.mean + X8.mean + X9.mean + X10.mean +
##     X11.mean + X12.mean + X13.mean + X1.sd + X2.sd + X3.sd +
##     X4.sd + X5.sd + X6.sd + X7.sd + X8.sd + X9.sd + X10.sd +
##     X11.sd + X12.sd + X13.sd, family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1381  -0.9970   0.5839   0.8396   1.9023
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.655235   2.780251   0.236  0.81369
## X1.mean     -0.021094   0.032545  -0.648  0.51690
## X2.mean      0.011411   0.041773   0.273  0.78473
## X3.mean     -0.022540   0.038818  -0.581  0.56147
## X4.mean     -0.030180   0.041469  -0.728  0.46676
## X5.mean     -0.004610   0.042610  -0.108  0.91384
## X6.mean     -0.032204   0.044280  -0.727  0.46706
## X7.mean     -0.088812   0.044350  -2.002  0.04523 *
## X8.mean      0.006439   0.043204   0.149  0.88152
## X9.mean      0.028228   0.049705   0.568  0.57010
## X10.mean    -0.052483   0.047284  -1.110  0.26703
## X11.mean     0.005584   0.045050   0.124  0.90135
## X12.mean     0.090631   0.048810   1.857  0.06334 .
## X13.mean     0.029204   0.053044   0.551  0.58194
## X1.sd       -0.047569   0.052317  -0.909  0.36322
## X2.sd       -0.102587   0.145505  -0.705  0.48079
## X3.sd       -0.260678   0.143443  -1.817  0.06917 .
## X4.sd        0.008038   0.171159   0.047  0.96254
## X5.sd        0.227364   0.171650   1.325  0.18531
## X6.sd        0.374222   0.171925   2.177  0.02951 *
## X7.sd       -0.180531   0.166870  -1.082  0.27931
## X8.sd        0.385520   0.209940   1.836  0.06631 .
## X9.sd        0.624075   0.224738   2.777  0.00549 **
## X10.sd       0.373502   0.207432   1.801  0.07177 .
## X11.sd      -0.653975   0.207134  -3.157  0.00159 **
## X12.sd      -0.095864   0.205302  -0.467  0.64054
## X13.sd      -0.309548   0.184964  -1.674  0.09422 .
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 455.53  on 367  degrees of freedom
## Residual deviance: 392.03  on 341  degrees of freedom
## AIC: 446.03
##
## Number of Fisher Scoring iterations: 5
```

```
##
## Call:
## glm(formula = y.train ~ X1.mean + X2.mean + X3.mean + X4.mean +
##     X5.mean + X6.mean + X7.mean + X8.mean + X9.mean + X10.mean +
##     X11.mean + X12.mean + X13.mean + X1.sd + X2.sd + X3.sd +
##     X4.sd + X5.sd + X6.sd + X7.sd + X8.sd + X9.sd + X10.sd +
##     X11.sd + X12.sd + X13.sd, family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2026  -0.9366   0.5714   0.8205   1.8622
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.8944096  5.9674920   0.485 0.627655
## X1.mean     -0.0252313  0.0327323  -0.771 0.440803
## X2.mean      0.0094736  0.0421803   0.225 0.822292
## X3.mean     -0.0207843  0.0388125  -0.536 0.592302
## X4.mean     -0.0250206  0.0416931  -0.600 0.548431
## X5.mean     -0.0002445  0.0425730  -0.006 0.995419
## X6.mean     -0.0350837  0.0441926  -0.794 0.427264
## X7.mean     -0.0995111  0.0445069  -2.236 0.025361 *
## X8.mean      0.0096768  0.0434388   0.223 0.823715
## X9.mean      0.0324635  0.0493442   0.658 0.510602
## X10.mean    -0.0586132  0.0467904  -1.253 0.210324
## X11.mean    -0.0016309  0.0450343  -0.036 0.971111
## X12.mean     0.0896375  0.0486701   1.842 0.065514 .
## X13.mean     0.0348536  0.0529199   0.659 0.510146
## X1.sd       -1.0015799  0.6843545  -1.464 0.143320
## X2.sd       -0.0618664  0.1443472  -0.429 0.668219
## X3.sd       -1.6181503  1.0025916  -1.614 0.106534
## X4.sd        0.0360625  1.2783788   0.028 0.977495
## X5.sd        1.8173815  1.2878676   1.411 0.158199
## X6.sd        2.3594241  1.1094427   2.127 0.033447 *
## X7.sd       -1.4284962  1.1567872  -1.235 0.216874
## X8.sd        2.6815262  1.2504759   2.144 0.032000 *
## X9.sd        3.9252286  1.3774240   2.850 0.004376 **
## X10.sd       2.2238859  1.2077630   1.841 0.065574 .
## X11.sd      -4.4375291  1.2978606  -3.419 0.000628 ***
## X12.sd      -3.5564393  5.5866878  -0.637 0.524391
## X13.sd      -1.7334947  1.1475409  -1.511 0.130886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 455.53  on 367  degrees of freedom
## Residual deviance: 389.55  on 341  degrees of freedom
## AIC: 443.55
##
## Number of Fisher Scoring iterations: 4
```

Training set misclassification error

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  35  79
##          1  17 237
##
##                Accuracy : 0.7391
##                  95% CI : (0.6911, 0.7833)
##     No Information Rate : 0.8587
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2824
##
##  Mcnemar's Test P-Value : 4.791e-10
##
##             Sensitivity : 0.67308
##             Specificity : 0.75000
##          Pos Pred Value : 0.30702
##          Neg Pred Value : 0.93307
##              Prevalence : 0.14130
##          Detection Rate : 0.09511
##    Detection Prevalence : 0.30978
##       Balanced Accuracy : 0.71154
##
##        'Positive' Class : 0
##
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  39  75
##          1  19 235
##
##                Accuracy : 0.7446
##                  95% CI : (0.6968, 0.7884)
##     No Information Rate : 0.8424
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3092
##
##  Mcnemar's Test P-Value : 1.405e-08
##
##             Sensitivity : 0.6724
```

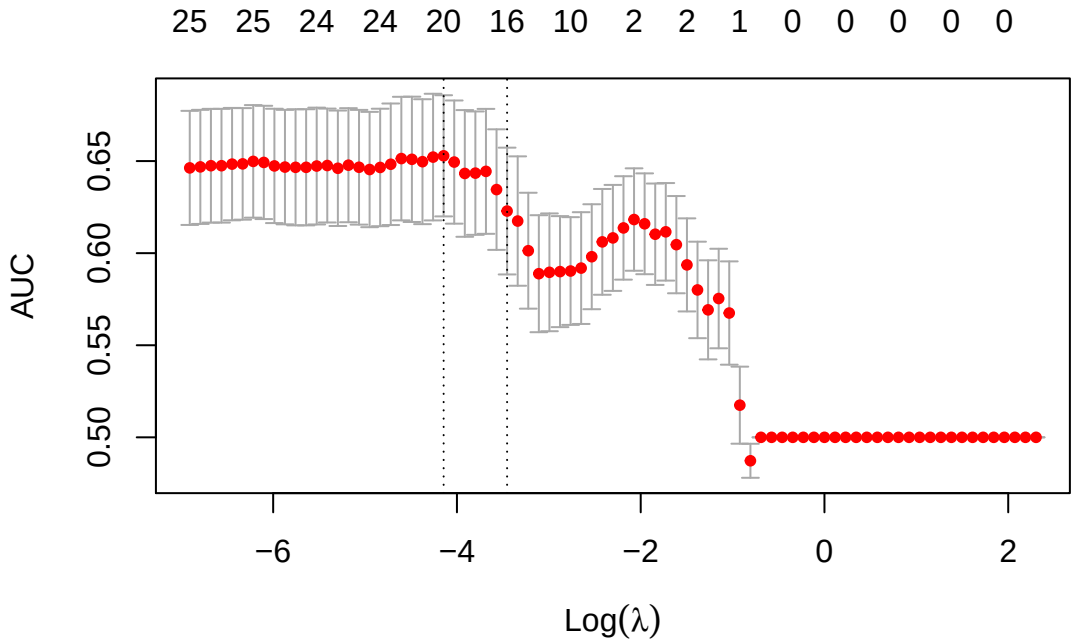
```
##             Specificity : 0.7581
##          Pos Pred Value : 0.3421
##          Neg Pred Value : 0.9252
##              Prevalence : 0.1576
##          Detection Rate : 0.1060
##    Detection Prevalence : 0.3098
##       Balanced Accuracy : 0.7152
##
##        'Positive' Class : 0
##
```

Performance on test set

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  9 21
##          1  5 53
##
##                Accuracy : 0.7045
##                  95% CI : (0.5978, 0.7971)
##     No Information Rate : 0.8409
##     P-Value [Acc > NIR] : 0.999601
##
##                   Kappa : 0.2454
##
##  Mcnemar's Test P-Value : 0.003264
##
##             Sensitivity : 0.6429
##             Specificity : 0.7162
##          Pos Pred Value : 0.3000
##          Neg Pred Value : 0.9138
##              Prevalence : 0.1591
##          Detection Rate : 0.1023
##    Detection Prevalence : 0.3409
##       Balanced Accuracy : 0.6795
##
##        'Positive' Class : 0
##
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  9 21
##          1  5 53
##
##                Accuracy : 0.7045
##                  95% CI : (0.5978, 0.7971)
##     No Information Rate : 0.8409
##     P-Value [Acc > NIR] : 0.999601
##
##                   Kappa : 0.2454
##
```

```
##  Mcnemar's Test P-Value : 0.003264
##
##             Sensitivity : 0.6429
##             Specificity : 0.7162
##          Pos Pred Value : 0.3000
##          Neg Pred Value : 0.9138
##              Prevalence : 0.1591
##          Detection Rate : 0.1023
##    Detection Prevalence : 0.3409
##       Balanced Accuracy : 0.6795
##
##        'Positive' Class : 0
##
```



LASSO selection

```
## [1] 0.01584893
```

LASSO best lambda

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0  57  57
##          1  37 217
##
##                Accuracy : 0.7446
##                  95% CI : (0.6968, 0.7884)
##     No Information Rate : 0.7446
##     P-Value [Acc > NIR] : 0.52769
```

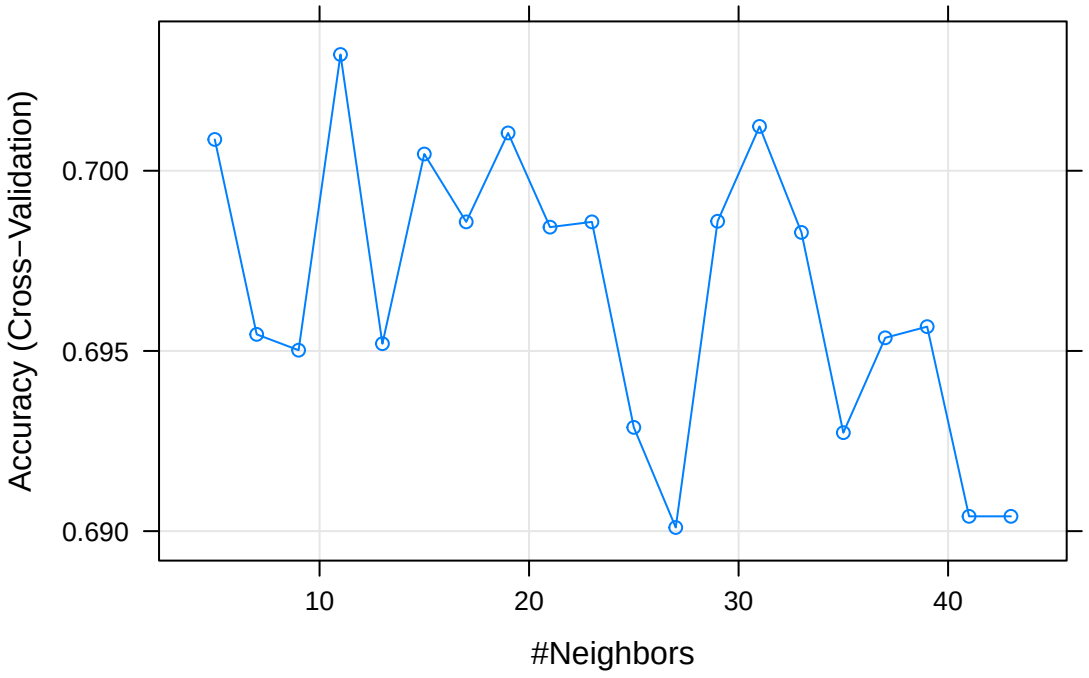
```
##
##                   Kappa : 0.3723
##
##  Mcnemar's Test P-Value : 0.05003
##
##             Sensitivity : 0.6064
##             Specificity : 0.7920
##          Pos Pred Value : 0.5000
##          Neg Pred Value : 0.8543
##              Prevalence : 0.2554
##          Detection Rate : 0.1549
##    Detection Prevalence : 0.3098
##       Balanced Accuracy : 0.6992
##
##        'Positive' Class : 0
##
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 14 16
##          1  9 49
##
##                Accuracy : 0.7159
##                  95% CI : (0.6098, 0.807)
##     No Information Rate : 0.7386
##     P-Value [Acc > NIR] : 0.7319
##
##                   Kappa : 0.3301
##
##  Mcnemar's Test P-Value : 0.2301
##
##             Sensitivity : 0.6087
##             Specificity : 0.7538
##          Pos Pred Value : 0.4667
##          Neg Pred Value : 0.8448
##              Prevalence : 0.2614
##          Detection Rate : 0.1591
##    Detection Prevalence : 0.3409
##       Balanced Accuracy : 0.6813
##
##        'Positive' Class : 0
##
```

KNN with K-Fold CV

```
## k-Nearest Neighbors
##
## 368 samples
##  26 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (26), scaled (26)
## Resampling: Cross-Validated (20 fold)
```

```
## Summary of sample sizes: 351, 349, 350, 350, 349, 350, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.7008686  0.20963115
##    7  0.6954592  0.15530588
##    9  0.6950206  0.10917919
##   11  0.7032250  0.11436675
##   13  0.6952012  0.09824216
##   15  0.7004644  0.09905287
##   17  0.6985810  0.08534996
##   19  0.7010492  0.09036321
##   21  0.6984348  0.06830981
##   23  0.6985810  0.06809232
##   25  0.6928793  0.04451819
##   27  0.6901015  0.02846990
##   29  0.6985982  0.04610567
##   31  0.7012298  0.04527923
##   33  0.6982886  0.03710545
##   35  0.6927331  0.02147710
##   37  0.6953646  0.02679662
##   39  0.6956742  0.02148760
##   41  0.6904111  0.00000000
##   43  0.6904111  0.00000000
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 11.
```




```
## 11-nearest neighbor model
## Training set outcome distribution:
```

```
##
##   0   1
## 114 254
```

```
## Cross-Validated (20 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction    0    1
##          0  4.3  3.0
##          1 26.6 66.0
##
##  Accuracy (average) : 0.7038
```

KNN on testing set

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0  4  2
##          1 26 56
##
##                Accuracy : 0.6818
##                  95% CI : (0.5739, 0.7771)
##     No Information Rate : 0.6591
##     P-Value [Acc > NIR] : 0.3721
##
##                   Kappa : 0.1225
##
##  Mcnemar's Test P-Value : 1.383e-05
##
##             Sensitivity : 0.13333
##             Specificity : 0.96552
##          Pos Pred Value : 0.66667
##          Neg Pred Value : 0.68293
##              Prevalence : 0.34091
##          Detection Rate : 0.04545
##    Detection Prevalence : 0.06818
##       Balanced Accuracy : 0.54943
##
##        'Positive' Class : 0
##
```

validation on best model

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 17 25
##          1 19 53
##
##                Accuracy : 0.614
```

```
##                  95% CI : (0.5183, 0.7037)
##     No Information Rate : 0.6842
##     P-Value [Acc > NIR] : 0.9547
##
##                   Kappa : 0.1452
##
##  Mcnemar's Test P-Value : 0.4510
##
##             Sensitivity : 0.4722
##             Specificity : 0.6795
##          Pos Pred Value : 0.4048
##          Neg Pred Value : 0.7361
##              Prevalence : 0.3158
##          Detection Rate : 0.1491
##    Detection Prevalence : 0.3684
##       Balanced Accuracy : 0.5759
##
##        'Positive' Class : 0
##
```

Kaggle dataset: predict gender

```
## [1] -0.02623901
```

```
## [1] -0.03820549
```

```
## [1] 0.5439599
```

```
## [1] 0.4020957
```

```
## [1] 0.4345694
```

```
## [1] 0.2190502
```

```
## [1] 0.1701115
```

```
## [1] 0.722784
```

```
## [1] 0.5910502
```

```
## [1] 0.5544879
```

```
## [1] 0.6339954
```

```
## [1] 0.7063097
```

```
## [1] 0.766185
```

Split Kaggle dataset: training, testing & validation sets

glm on training data

```
##
## Call:
## glm(formula = y.train ~ X1.mean + X2.mean + X3.mean + X4.mean +
##     X5.mean + X6.mean + X7.mean + X8.mean + X9.mean + X10.mean +
##     X11.mean + X12.mean + X13.mean + X1.sd + X2.sd + X3.sd +
##     X4.sd + X5.sd + X6.sd + X7.sd + X8.sd + X9.sd + X10.sd +
##     X11.sd + X12.sd + X13.sd, family = binomial)
##
```

```
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.82253  -0.51146   0.07435   0.50251   2.77453
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.86028    1.70268   2.267 0.023379 *
## X1.mean     -0.02621    0.01799  -1.457 0.145024
## X2.mean     -0.06307    0.02720  -2.319 0.020399 *
## X3.mean     -0.18153    0.02690  -6.748 1.50e-11 ***
## X4.mean     -0.09664    0.02882  -3.353 0.000800 ***
## X5.mean      0.20611    0.03025   6.813 9.57e-12 ***
## X6.mean      0.26511    0.02850   9.302  < 2e-16 ***
## X7.mean      0.20940    0.03093   6.770 1.29e-11 ***
## X8.mean      0.06924    0.02768   2.502 0.012350 *
## X9.mean      0.06511    0.02963   2.198 0.027981 *
## X10.mean    -0.02451    0.03227  -0.759 0.447607
## X11.mean    -0.07823    0.03005  -2.603 0.009231 **
## X12.mean    -0.04035    0.03257  -1.239 0.215382
## X13.mean     0.16758    0.03316   5.053 4.35e-07 ***
## X1.sd        0.07681    0.03537   2.172 0.029888 *
## X2.sd       -0.07280    0.09442  -0.771 0.440683
## X3.sd       -0.07470    0.10214  -0.731 0.464545
## X4.sd       -0.09225    0.10569  -0.873 0.382732
## X5.sd       -0.17391    0.11013  -1.579 0.114327
## X6.sd        0.93808    0.11771   7.970 1.59e-15 ***
## X7.sd       -0.32378    0.09495  -3.410 0.000649 ***
## X8.sd       -0.74810    0.13440  -5.566 2.61e-08 ***
## X9.sd        0.19638    0.12575   1.562 0.118373
## X10.sd      -0.57237    0.12978  -4.410 1.03e-05 ***
## X11.sd      -0.14664    0.13063  -1.123 0.261643
## X12.sd      -1.16538    0.13550  -8.601  < 2e-16 ***
## X13.sd       1.96316    0.15224  12.895  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1861.74  on 1343  degrees of freedom
## Residual deviance:  967.36  on 1317  degrees of freedom
## AIC: 1021.4
##
## Number of Fisher Scoring iterations: 6
##
## Call:
## glm(formula = y.train ~ X1.mean + X2.mean + X3.mean + X4.mean +
##     X5.mean + X6.mean + X7.mean + X8.mean + X9.mean + X10.mean +
##     X11.mean + X12.mean + X13.mean + X1.sd + X2.sd + X3.sd +
##     X4.sd + X5.sd + X6.sd + X7.sd + X8.sd + X9.sd + X10.sd +
##     X11.sd + X12.sd + X13.sd, family = binomial)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
```

```
## -2.97281  -0.49471   0.08195   0.49277   2.83951
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  32.08798    3.90733   8.212  < 2e-16 ***
## X1.mean      -0.02797    0.01775  -1.576 0.115109
## X2.mean      -0.05749    0.02752  -2.089 0.036666 *
## X3.mean      -0.18252    0.02686  -6.795 1.08e-11 ***
## X4.mean      -0.09498    0.02894  -3.282 0.001029 **
## X5.mean       0.20737    0.03069   6.758 1.40e-11 ***
## X6.mean       0.26782    0.02884   9.287  < 2e-16 ***
## X7.mean       0.21142    0.03114   6.789 1.13e-11 ***
## X8.mean       0.06611    0.02775   2.382 0.017199 *
## X9.mean       0.06477    0.02952   2.194 0.028242 *
## X10.mean     -0.02579    0.03220  -0.801 0.423124
## X11.mean     -0.09328    0.02995  -3.114 0.001843 **
## X12.mean     -0.03369    0.03258  -1.034 0.300964
## X13.mean      0.17528    0.03363   5.212 1.87e-07 ***
## X1.sd         1.17006    0.48334   2.421 0.015486 *
## X2.sd        -0.55241    0.82677  -0.668 0.504036
## X3.sd        -0.84017    0.68127  -1.233 0.217485
## X4.sd        -0.85171    0.79782  -1.068 0.285723
## X5.sd        -1.44360    0.83498  -1.729 0.083826 .
## X6.sd         6.33752    0.76042   8.334  < 2e-16 ***
## X7.sd        -2.22056    0.65593  -3.385 0.000711 ***
## X8.sd        -4.45337    0.79883  -5.575 2.48e-08 ***
## X9.sd         1.41381    0.78766   1.795 0.072663 .
## X10.sd       -3.33143    0.77137  -4.319 1.57e-05 ***
## X11.sd       -1.45134    0.79260  -1.831 0.067083 .
## X12.sd      -32.67943    3.74148  -8.734  < 2e-16 ***
## X13.sd       11.80467    0.89680  13.163  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1861.74  on 1343  degrees of freedom
## Residual deviance:  945.98  on 1317  degrees of freedom
## AIC: 999.98
##
## Number of Fisher Scoring iterations: 6
```

Training set misclassification error

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 538 112
##          1 112 582
##
##                Accuracy : 0.8333
##                  95% CI : (0.8123, 0.8529)
##     No Information Rate : 0.5164
```

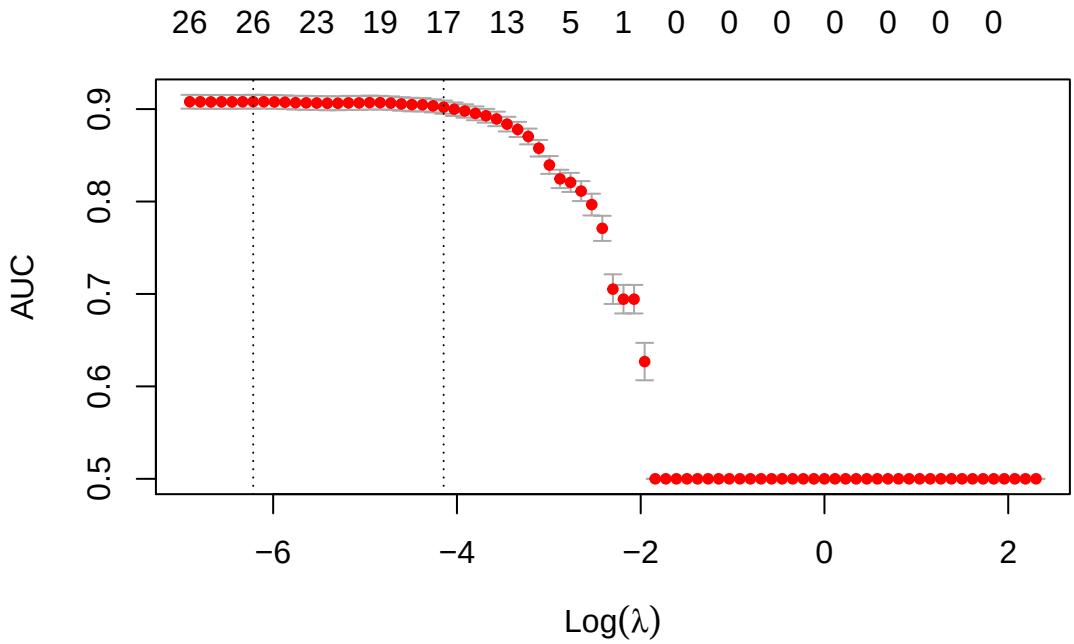
```
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6663
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8277
##             Specificity : 0.8386
##          Pos Pred Value : 0.8277
##          Neg Pred Value : 0.8386
##              Prevalence : 0.4836
##          Detection Rate : 0.4003
##    Detection Prevalence : 0.4836
##       Balanced Accuracy : 0.8332
##
##        'Positive' Class : 0
##
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 542 103
##          1 108 591
##
##                Accuracy : 0.843
##                  95% CI : (0.8224, 0.8621)
##     No Information Rate : 0.5164
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6856
##
##  Mcnemar's Test P-Value : 0.783
##
##             Sensitivity : 0.8338
##             Specificity : 0.8516
##          Pos Pred Value : 0.8403
##          Neg Pred Value : 0.8455
##              Prevalence : 0.4836
##          Detection Rate : 0.4033
##    Detection Prevalence : 0.4799
##       Balanced Accuracy : 0.8427
##
##        'Positive' Class : 0
##
```

Performance on test set

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 133  27
##          1  29 143
##
```

```
##                Accuracy : 0.8313
##                  95% CI : (0.7866, 0.87)
##     No Information Rate : 0.512
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6624
##
##  Mcnemar's Test P-Value : 0.8937
##
##             Sensitivity : 0.8210
##             Specificity : 0.8412
##          Pos Pred Value : 0.8312
##          Neg Pred Value : 0.8314
##              Prevalence : 0.4880
##          Detection Rate : 0.4006
##    Detection Prevalence : 0.4819
##       Balanced Accuracy : 0.8311
##
##        'Positive' Class : 0
##
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 134  28
##          1  28 142
##
##                Accuracy : 0.8313
##                  95% CI : (0.7866, 0.87)
##     No Information Rate : 0.512
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6625
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8272
##             Specificity : 0.8353
##          Pos Pred Value : 0.8272
##          Neg Pred Value : 0.8353
##              Prevalence : 0.4880
##          Detection Rate : 0.4036
##    Detection Prevalence : 0.4880
##       Balanced Accuracy : 0.8312
##
##        'Positive' Class : 0
##
```



LASSO selection

```
## [1] 0.001995262
```

LASSO best lambda

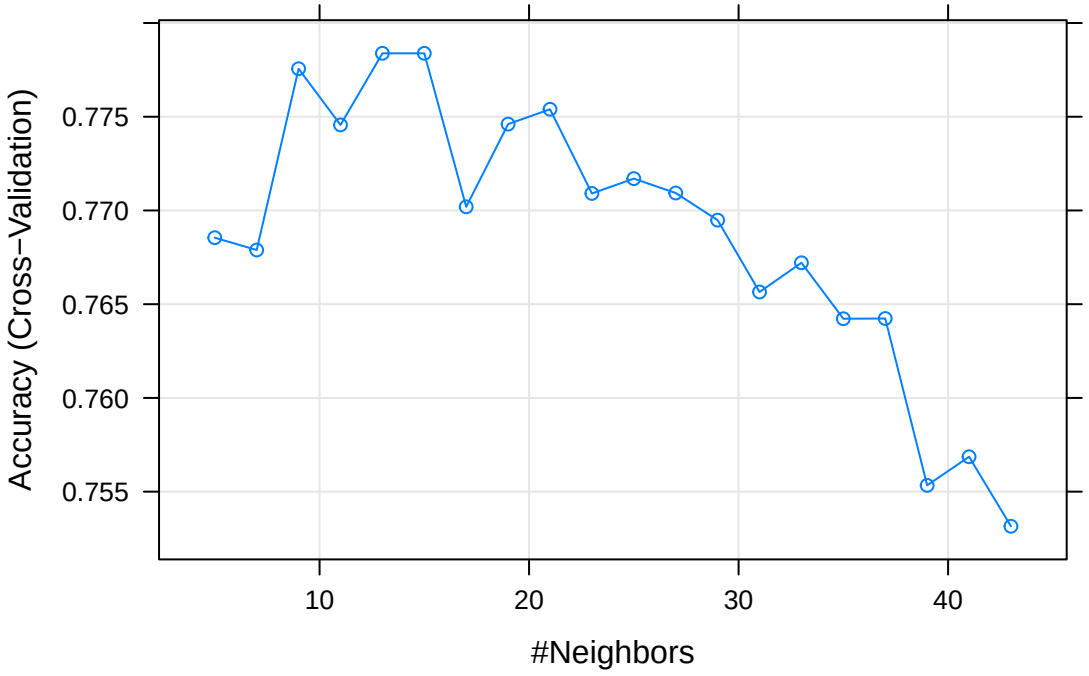
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 576 163
##          1  74 531
##
##                Accuracy : 0.8237
##                  95% CI : (0.8022, 0.8437)
##     No Information Rate : 0.5164
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6485
##
##  Mcnemar's Test P-Value : 1.089e-08
##
##             Sensitivity : 0.8862
##             Specificity : 0.7651
##          Pos Pred Value : 0.7794
##          Neg Pred Value : 0.8777
##              Prevalence : 0.4836
##          Detection Rate : 0.4286
##    Detection Prevalence : 0.5499
##       Balanced Accuracy : 0.8256
##
##        'Positive' Class : 0
```

```
##
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 140  41
##          1  22 129
##
##                Accuracy : 0.8102
##                  95% CI : (0.7639, 0.851)
##     No Information Rate : 0.512
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6213
##
##  Mcnemar's Test P-Value : 0.02334
##
##             Sensitivity : 0.8642
##             Specificity : 0.7588
##          Pos Pred Value : 0.7735
##          Neg Pred Value : 0.8543
##              Prevalence : 0.4880
##          Detection Rate : 0.4217
##    Detection Prevalence : 0.5452
##       Balanced Accuracy : 0.8115
##
##        'Positive' Class : 0
##
```

KNN with K-Fold CV

```
## k-Nearest Neighbors
##
## 1344 samples
##   26 predictor
##    2 classes: '0', '1'
##
## Pre-processing: centered (26), scaled (26)
## Resampling: Cross-Validated (20 fold)
## Summary of sample sizes: 1277, 1277, 1276, 1276, 1276, 1277, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.7685476  0.5373877
##    7  0.7678905  0.5356315
##    9  0.7775604  0.5553791
##   11  0.7745633  0.5493010
##   13  0.7783838  0.5568994
##   15  0.7783832  0.5567415
##   17  0.7701958  0.5402914
##   19  0.7746076  0.5490393
##   21  0.7753977  0.5505663
##   23  0.7709088  0.5419636
##   25  0.7716987  0.5435031
```

```
##   27  0.7709308  0.5416595
##   29  0.7694831  0.5387550
##   31  0.7656514  0.5313431
##   33  0.7672097  0.5345291
##   35  0.7642250  0.5286404
##   37  0.7642363  0.5285495
##   39  0.7553363  0.5106779
##   41  0.7568617  0.5136390
##   43  0.7531524  0.5064321
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 13.
```




```
## 13-nearest neighbor model
## Training set outcome distribution:
##
##   0   1
## 650 694
```

```
## Cross-Validated (20 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##            Reference
## Prediction    0    1
##          0 38.5 12.3
##          1  9.9 39.4
##
##  Accuracy (average) : 0.7783
```

KNN on testing set

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 126  41
##          1  36 129
##
##                Accuracy : 0.7681
##                  95% CI : (0.7189, 0.8124)
##     No Information Rate : 0.512
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.5362
##
##  Mcnemar's Test P-Value : 0.6485
##
##             Sensitivity : 0.7778
##             Specificity : 0.7588
##          Pos Pred Value : 0.7545
##          Neg Pred Value : 0.7818
##              Prevalence : 0.4880
##          Detection Rate : 0.3795
##    Detection Prevalence : 0.5030
##       Balanced Accuracy : 0.7683
##
##        'Positive' Class : 0
##
```

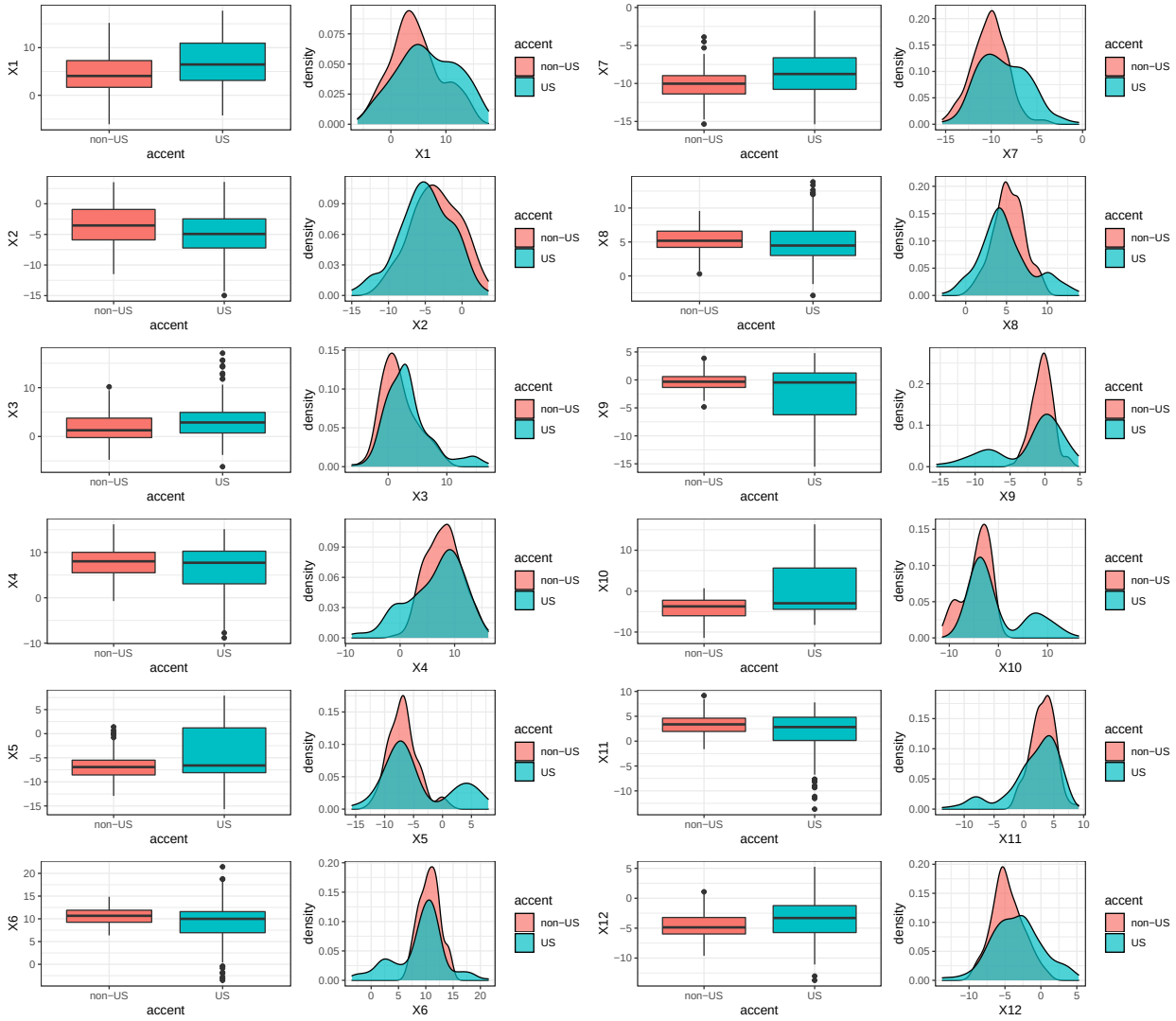
validation on best model

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 162  30
##          1  40 185
##
##                Accuracy : 0.8321
##                  95% CI : (0.7927, 0.8667)
##     No Information Rate : 0.5156
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6634
##
##  Mcnemar's Test P-Value : 0.2821
##
##             Sensitivity : 0.8020
##             Specificity : 0.8605
##          Pos Pred Value : 0.8438
##          Neg Pred Value : 0.8222
##              Prevalence : 0.4844
##          Detection Rate : 0.3885
##    Detection Prevalence : 0.4604
```

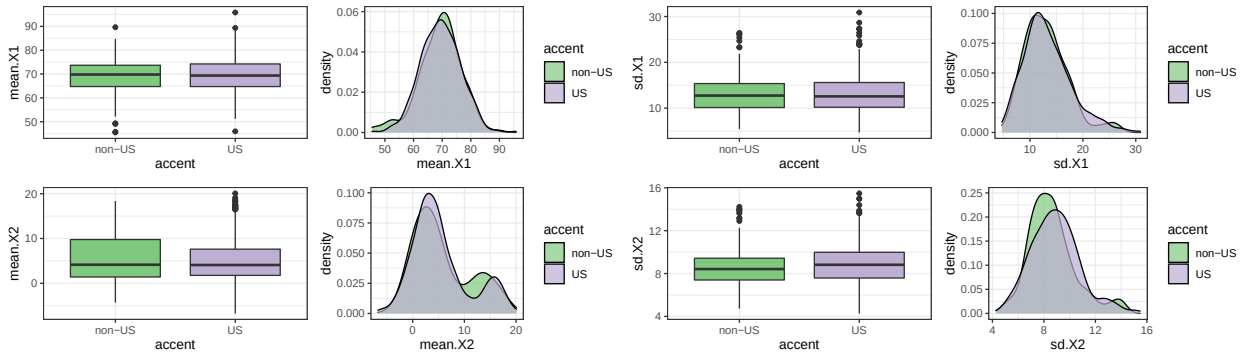
```
##       Balanced Accuracy : 0.8312
##
##        'Positive' Class : 0
##
```

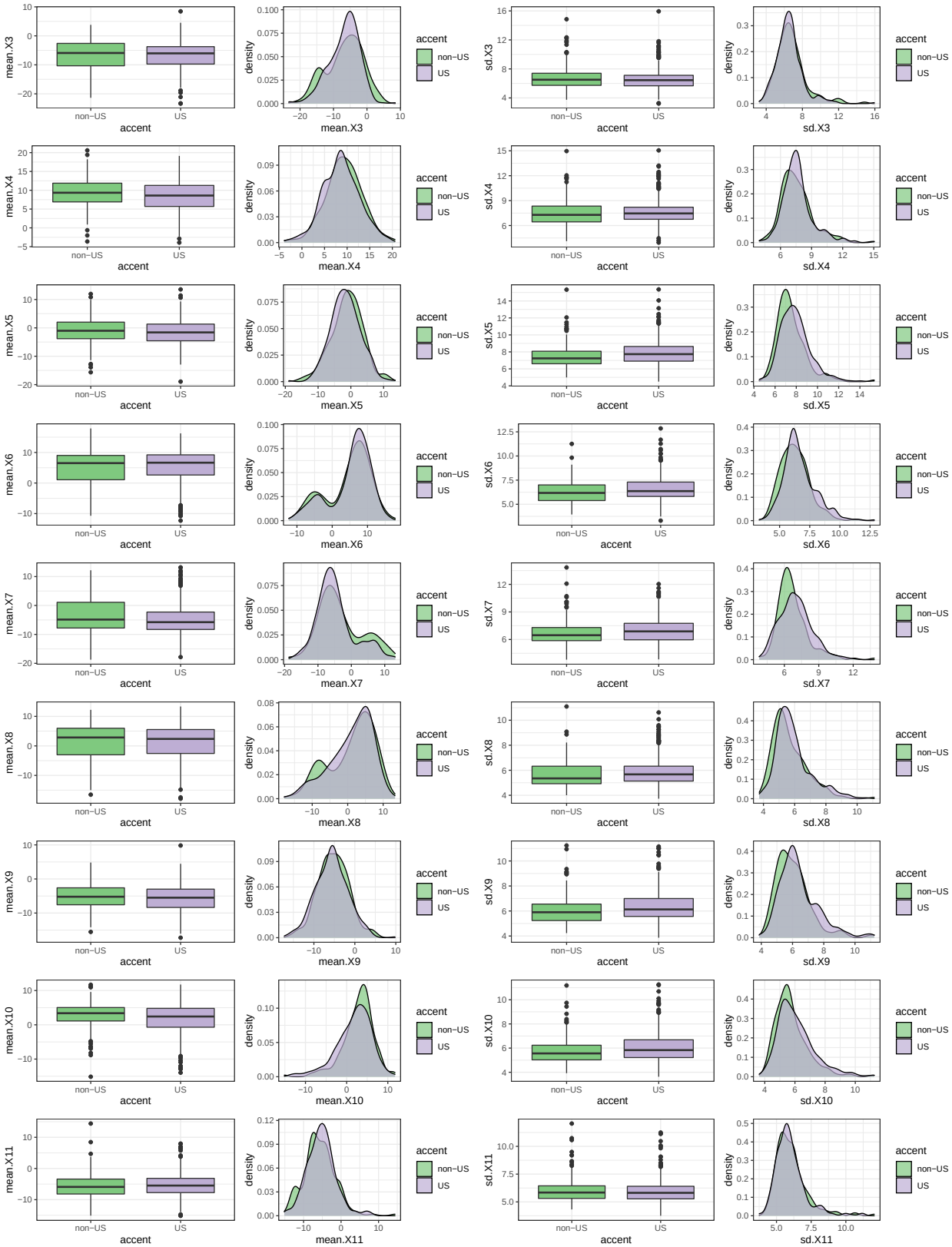
Appendix 2: Related Plots

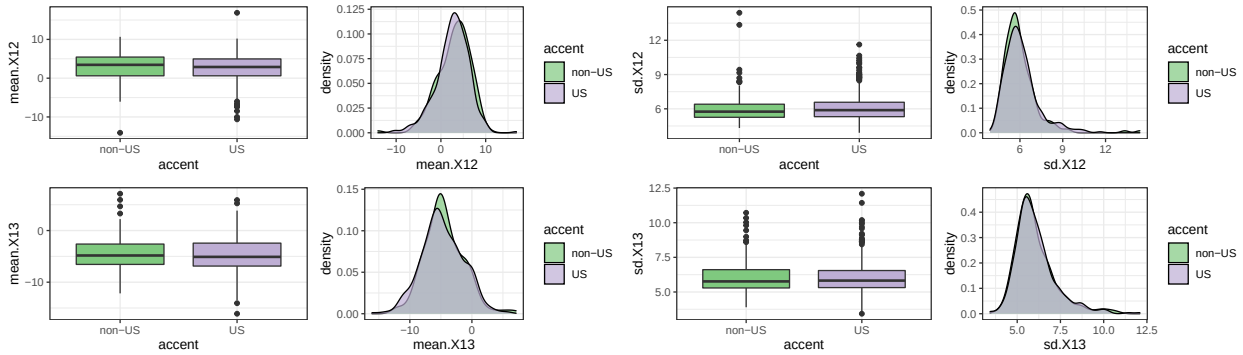
UCI dataset: Features X1,. . . ,X12; Response accent



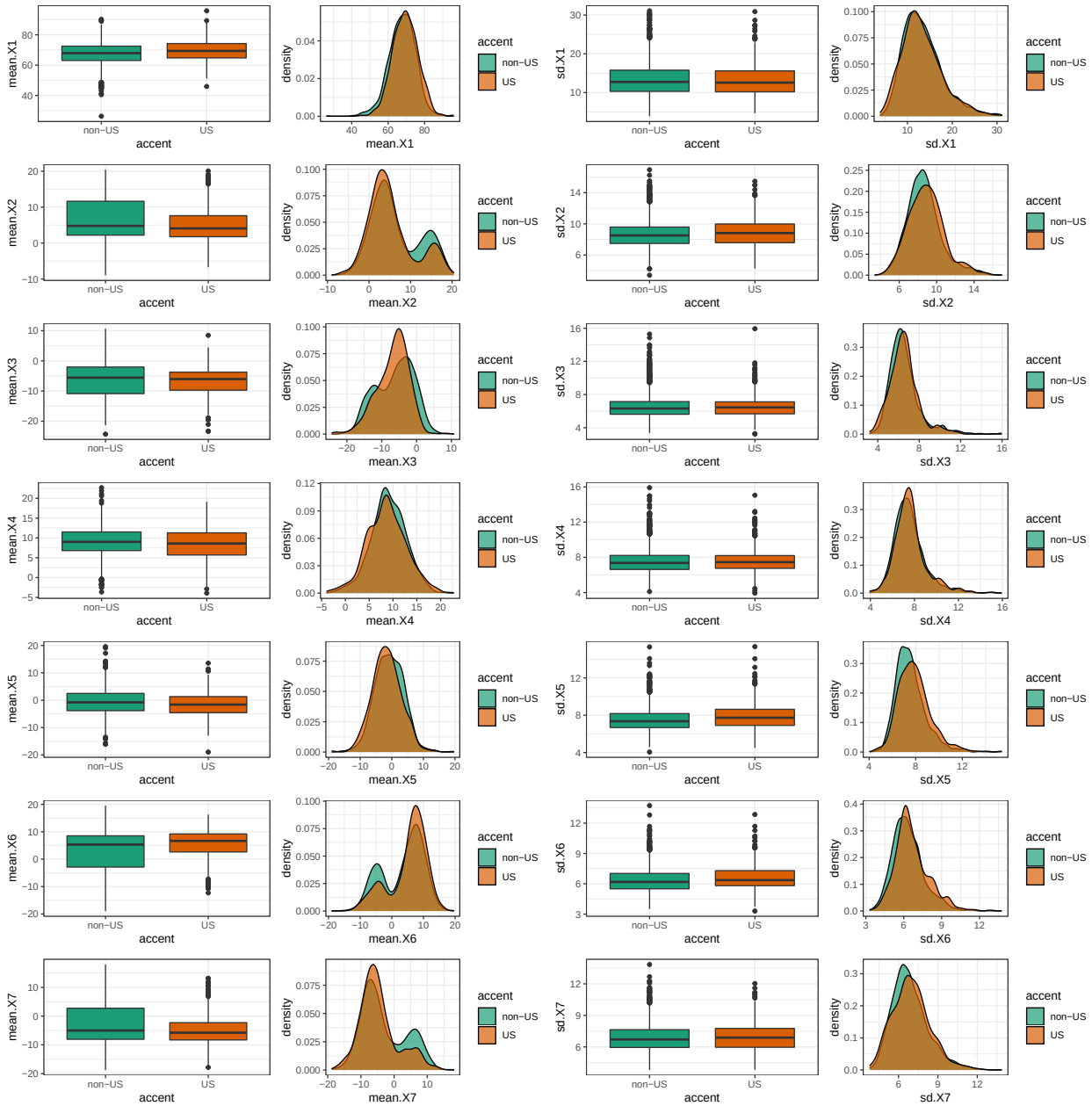
Kaggle dataset (filtered): Features: X1,. . . ,X12 (equivalent to UCI dataset, including MFCC #2 to #13); response: accent (equivalent to UCI dataset, which only has {ES, FR, GE, IT, UK, US})

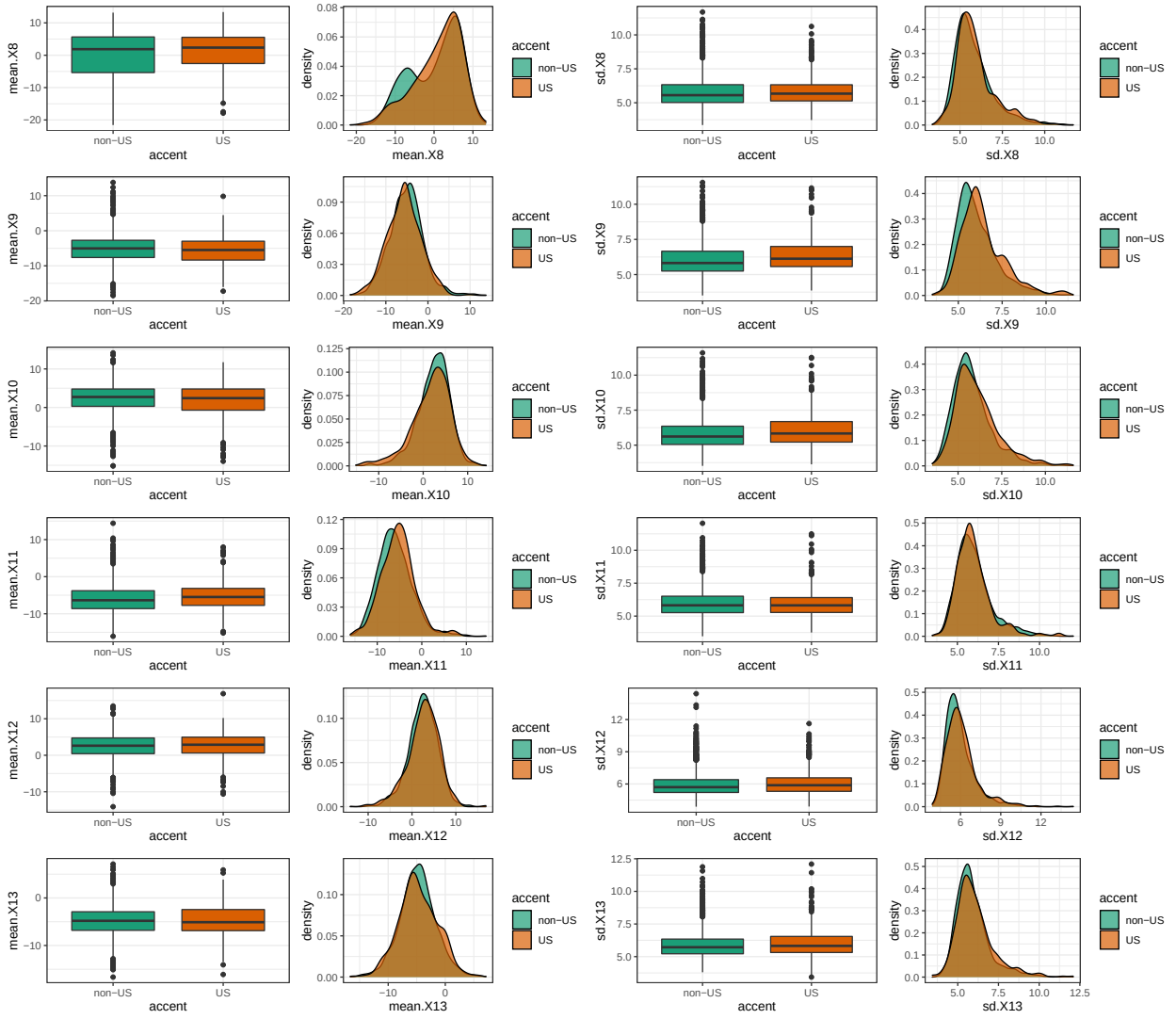



mean.X3
density
sd.X3
mean.X4
sd.X4
mean.X5
sd.X5
mean.X6
sd.X6
mean.X7
sd.X7
mean.X8
sd.X8
mean.X9
sd.X9
mean.X10
sd.X10
mean.X11
sd.X11
accent
non−US
US

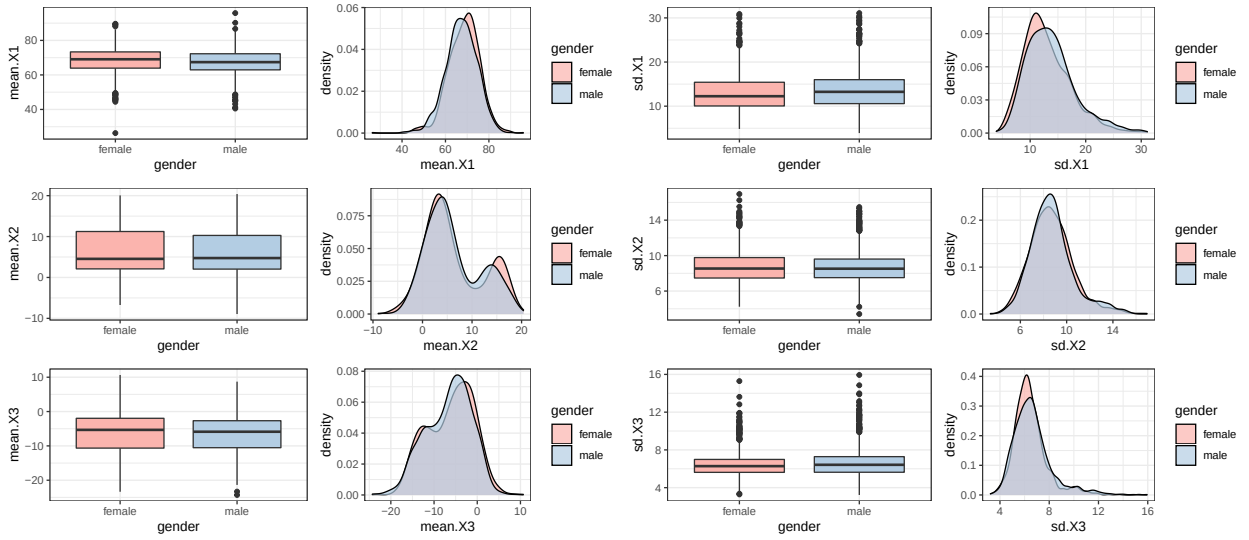


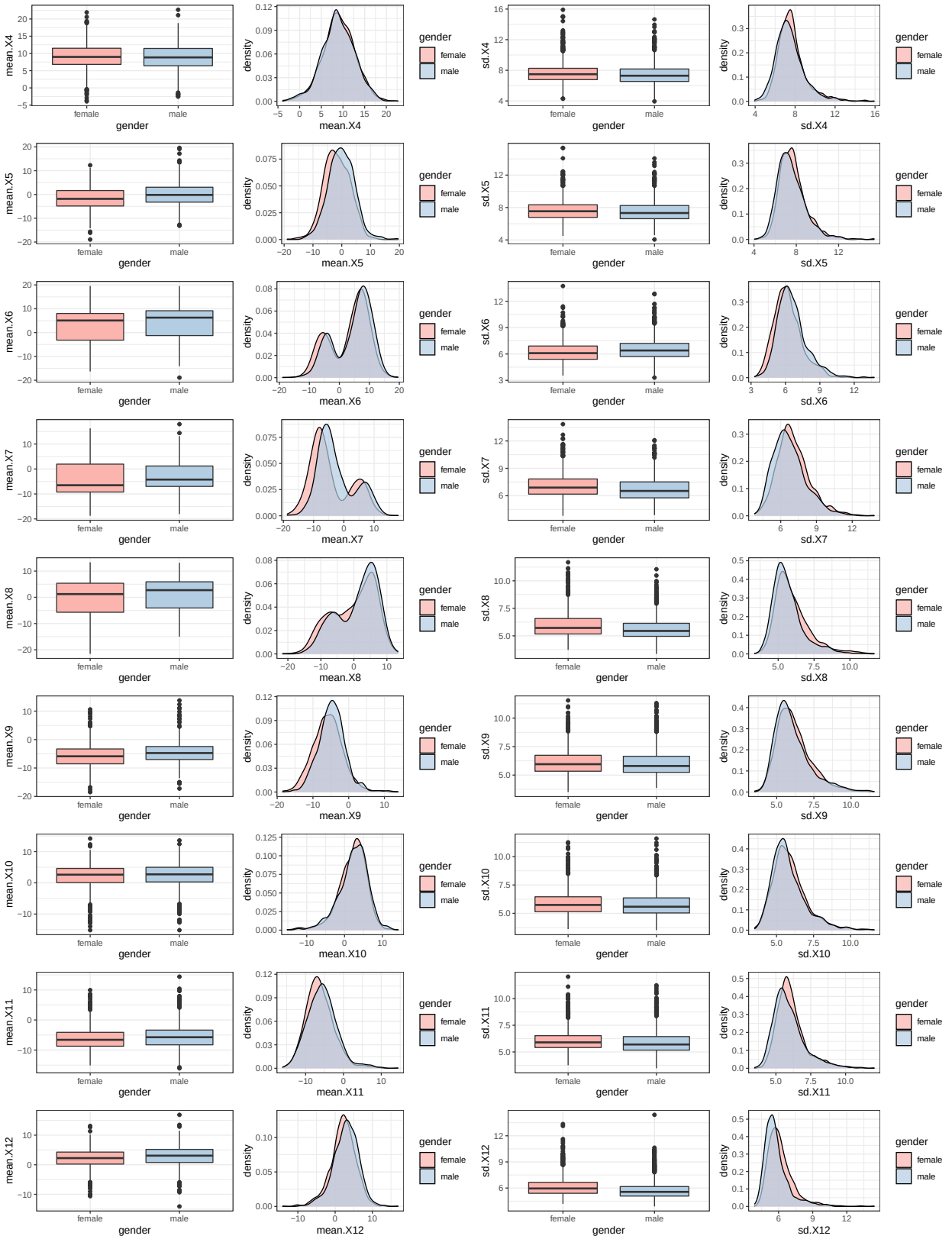
Kaggle dataset (full): Features: means & standard deviations of X1,...,X13: Response: accent





Kaggle dataset (full): Features: means & standard deviations of X1,...,X13: Response: gender




mean.X4
mean.X5
mean.X6
mean.X7
mean.X8
mean.X9
mean.X10
mean.X11
mean.X12
sd.X4
sd.X5
sd.X6
sd.X7
sd.X8
sd.X9
sd.X10
sd.X11
sd.X12
density
gender
female
male


mean.X13
0
−10
female
male
gender
density
0.10
0.05
0.00
−10
0
mean.X13
gender
female
male
sd.X13
12.5
10.0
7.5
5.0
female
male
gender
density
0.6
0.4
0.2
0.0
5.0
7.5
10.0
12.5
sd.X13
gender
female
male