SPAM Classification

Ideas to use (SVM, multinomial naïve bayes… GAM?, decision trees)

http://topepo.github.io/caret/index.html

Purpose:

Our purpose is to build a spam detection engine. This is a classification problem as the outcome should be either 0 for no spam and 1 for spam.

Data for Model:

- We get our dataset from the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Spambase
- This dataset contains real email examples with labeled spam attributes.
- 4601 records in the dataset (1813 Spam = 39.4%)
- Class Distribution:
    - Spam: 1813 (39.4%)
    - Non-Spam: 2788 (60.6%)
- The dataset contains 57 attributes or features.
- The last column of the data file denotes whether the e-mail was considered spam (1) or not (0)
- Most of the attributes indicate whether a word or character was frequently occurring in the e-mail.
- The run-length attributes (55-57) measure the length of sequences of consecutive capital letters.
  Here are the definitions of the attributes:
    - **48** continuous real [0,100] attributes of type word_freq_WORD
      = percentage of words in the e-mail that match WORD, i.e. 100 * (number of times the WORD appears in the e-mail) / total number of words in e-mail. A "word" in this case is any string of alphanumeric characters bounded by non-alphanumeric characters or end-of-string.

    - **6** continuous real [0,100] attributes of type char_freq_CHAR]
      = percentage of characters in the e-mail that match CHAR, i.e. 100 * (number of CHAR occurences) / total characters in e-mail

    - **1** continuous real [1,...] attribute of type capital_run_length_average
      = average length of uninterrupted sequences of capital letters

    - **1** continuous integer [1,...] attribute of type capital_run_length_longest
      = length of longest uninterrupted sequence of capital letters

    - **1** continuous integer [1,...] attribute of type capital_run_length_total
      = sum of length of uninterrupted sequences of capital letters
      = total number of capital letters in the e-mail

- **1** nominal {0,1} class attribute of type spam
  = denotes whether the e-mail was considered spam (1) or not (0), i.e. unsolicited commercial e-mail.

Models:

1. ***Model 1: SVM Classifier***
   - *An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.*
   - In, general the SVM algorithm tries to **separate the data with a gap that is as wide as possible**. It does so **with the help of vectors which define hyperplanes**.

Feature Selection – Not Necessary

The SVM is an approximate implementation of a theoretical bound on the generalization performance that is independent of the dimensionality of the feature space. This means that there is a good reason to suggest that performing feature selection might not make the performance of the classifier any better.

The reason that the SVM works is because it uses regularization (like ridge regression) to avoid over-fitting, so provided you set the regularization parameter C properly (e.g. using cross-validation), the performance ought to be good without feature selection.

The thing that is often not mentioned about feature selection is that it can easily make performance *worse*. The reason for this is that the more choices about the model that are made by optimizing some statistic evaluated over the training sample, the more likely you are to over-fit the training sample, and feature selection often ends up making many more choices about the model.

Tuning Parameter C

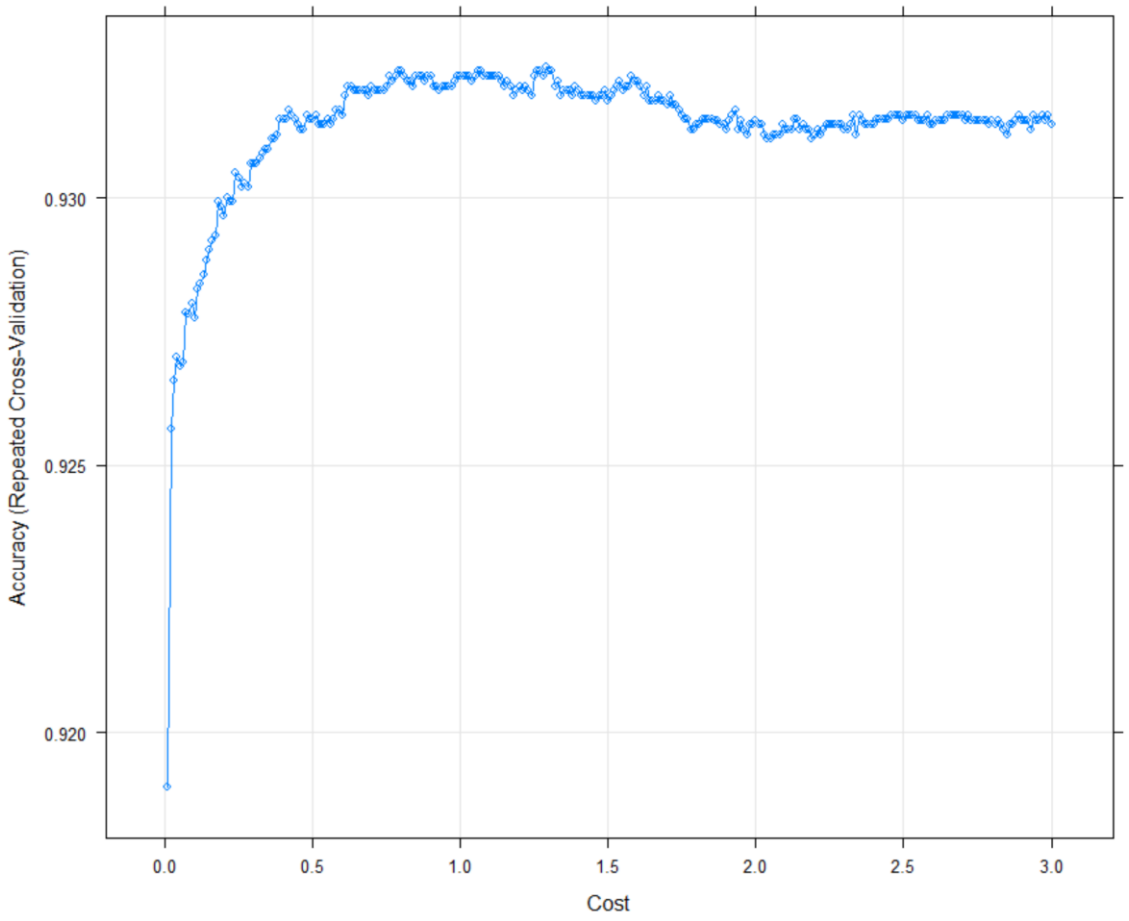
The C parameter tells the SVM optimization how much you want to avoid misclassifying each training example.

For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly.

Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points. For very tiny values of C, you should get misclassified examples, often even if your training data is linearly separable.

1. ***Linear SVM Model:***
   - Used grid to find optimal c value.
   - 10-fold CV, repeated 3 times.
   - Multi-core processing using caret package.
   - Best C value = 1.29
   - Accuracy = 93%

Tuning Values for C as a function of Accuracy (Repeated CV)



Test Set Results:

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 527 40
         1 30 322

               Accuracy: 0.9238
                 95% CI: (0.9047, 0.9401)
    No Information Rate: 0.6061
    P-Value [Acc > NIR]: <2e-16

                  Kappa: 0.8397
 Mcnemar's Test P-Value: 0.2821
```

```
          Sensitivity: 0.9461
          Specificity: 0.8895
       Pos Pred Value: 0.9295
       Neg Pred Value: 0.9148
           Prevalence: 0.6061
       Detection Rate: 0.5734
 Detection Prevalence: 0.6170
    Balanced Accuracy: 0.9178

     'Positive' Class: 0
```

2. Radial SVM Model:

- Use grid to find optimal Cost C parameter and sigma parameter.
- Will run every value of sigma for each value of C.
- Must retune and adjust values repeatedly.

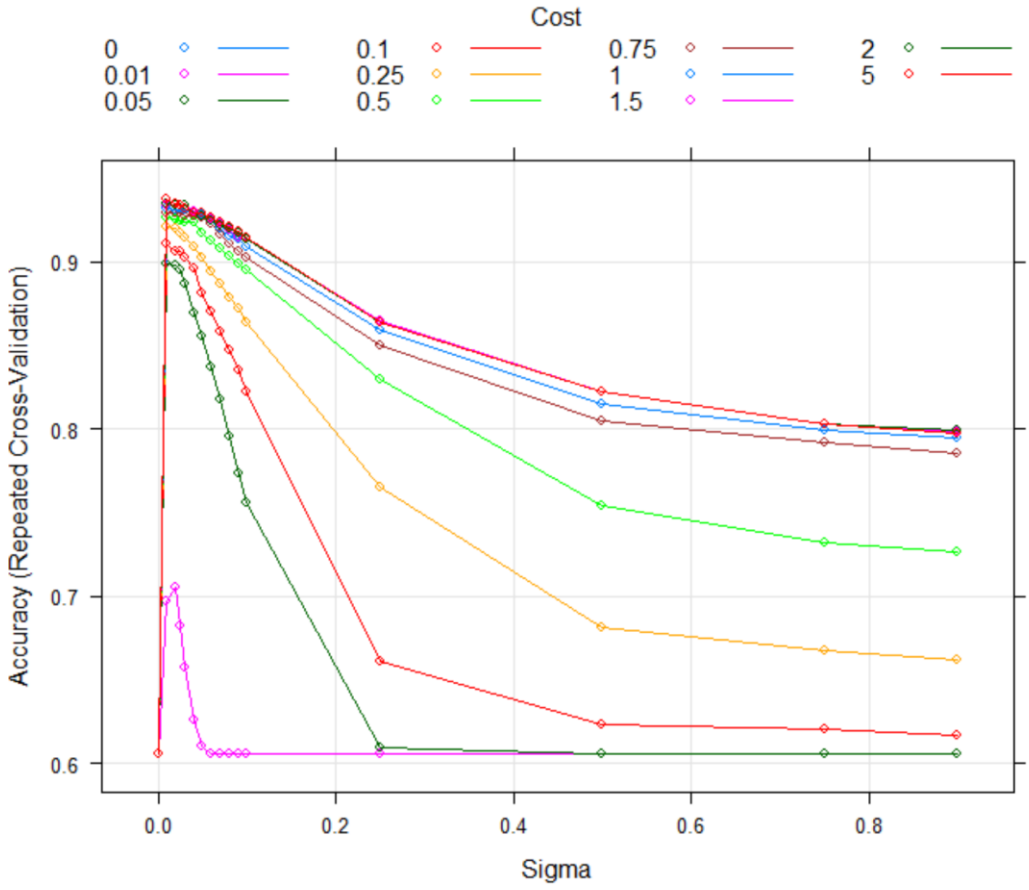
```
• Support Vector Machines with Radial Basis Function Kernel
•
• 3682 samples
•   57 predictor
•    2 classes: '0', '1'
•
• Pre-processing: centered (57), scaled (57)
• Resampling: Cross-Validated (10 fold, repeated 3 times)
• Summary of sample sizes: 3314, 3314, 3313, 3314, 3314, 3314, ...
• Resampling results across tuning parameters:
•
•   sigma  C     Accuracy   Kappa
•   0.000  0.00        NaN         NaN
•   0.000  0.01  0.6059208  0.000000000
•   0.000  0.05  0.6059208  0.000000000
•   0.000  0.10  0.6059208  0.000000000
•   0.000  0.25  0.6059208  0.000000000
•   0.000  0.50  0.6059208  0.000000000
•   0.000  0.75  0.6059208  0.000000000
•   0.000  1.00  0.6059208  0.000000000
•   0.000  1.50  0.6059208  0.000000000
•   0.000  2.00  0.6059208  0.000000000
•   0.000  5.00  0.6059208  0.000000000
•   0.010  0.00        NaN         NaN
•   0.010  0.01  0.6965408  0.266040073
•   0.010  0.05  0.8989666  0.782054790
•   0.010  0.10  0.9104638  0.808389420
•   0.010  0.25  0.9207837  0.831477534
•   0.010  0.50  0.9263076  0.843764362
•   0.010  0.75  0.9296569  0.851093156
•   0.010  1.00  0.9322825  0.856893190
•   0.010  1.50  0.9339114  0.860504254
•   0.010  2.00  0.9349986  0.862837445
```

•	0.010	5.00	0.9377138	0.868799255
•	0.020	0.00	NaN	NaN
•	0.020	0.01	0.7052291	0.290888938
•	0.020	0.05	0.8976979	0.779411275
•	0.020	0.10	0.9063904	0.799319694
•	0.020	0.25	0.9200606	0.829937056
•	0.020	0.50	0.9253113	0.841553948
•	0.020	0.75	0.9289342	0.849584276
•	0.020	1.00	0.9308361	0.853859377
•	0.020	1.50	0.9338235	0.860305508
•	0.020	2.00	0.9340941	0.860955470
•	0.020	5.00	0.9346363	0.862109856
•	0.025	0.00	NaN	NaN
•	0.025	0.01	0.6816916	0.224952554
•	0.025	0.05	0.8956161	0.774486538
•	0.025	0.10	0.9058462	0.798050099
•	0.025	0.25	0.9175263	0.824213558
•	0.025	0.50	0.9243154	0.839431640
•	0.025	0.75	0.9289340	0.849545580
•	0.025	1.00	0.9317414	0.855754803
•	0.025	1.50	0.9334602	0.859489693
•	0.025	2.00	0.9340928	0.860820841
•	0.025	5.00	0.9338213	0.860356014
•	0.030	0.00	NaN	NaN
•	0.030	0.01	0.6574322	0.155256621
•	0.030	0.05	0.8871060	0.755023897
•	0.030	0.10	0.9024972	0.790387324
•	0.030	0.25	0.9147179	0.817989771
•	0.030	0.50	0.9239535	0.838522238
•	0.030	0.75	0.9276681	0.846814521
•	0.030	1.00	0.9307448	0.853510914
•	0.030	1.50	0.9340027	0.860538343
•	0.030	2.00	0.9336397	0.859857601
•	0.030	5.00	0.9317385	0.855838708
•	0.040	0.00	NaN	NaN
•	0.040	0.01	0.6259281	0.061079587
•	0.040	0.05	0.8698107	0.714878572
•	0.040	0.10	0.8957054	0.774491911
•	0.040	0.25	0.9093768	0.805826227
•	0.040	0.50	0.9233215	0.837004600
•	0.040	0.75	0.9275755	0.846501273
•	0.040	1.00	0.9290228	0.849664876
•	0.040	1.50	0.9304716	0.852887867
•	0.040	2.00	0.9296561	0.851213800
•	0.040	5.00	0.9291141	0.849811266
•	0.050	0.00	NaN	NaN
•	0.050	0.01	0.6105382	0.014434580
•	0.050	0.05	0.8556005	0.681448717
•	0.050	0.10	0.8813995	0.741600492

```
•   0.050  0.25  0.9021346  0.789439141
•   0.050  0.50  0.9175266  0.824016626
•   0.050  0.75  0.9263980  0.843639231
•   0.050  1.00  0.9282088  0.847736021
•   0.050  1.50  0.9292946  0.850242959
•   0.050  2.00  0.9273939  0.846176153
•   0.050  5.00  0.9289315  0.849260905
•   0.060  0.00        NaN          NaN
•   0.060  0.01  0.6060114  0.000278204
•   0.060  0.05  0.8371330  0.637017426
•   0.060  0.10  0.8707175  0.716837751
•   0.060  0.25  0.8946204  0.772262496
•   0.060  0.50  0.9127298  0.813277740
•   0.060  0.75  0.9226871  0.835433791
•   0.060  1.00  0.9247693  0.840226834
•   0.060  1.50  0.9258545  0.842656142
•   0.060  2.00  0.9251303  0.840938566
•   0.060  5.00  0.9268489  0.844634844
•   0.070  0.00        NaN          NaN
•   0.070  0.01  0.6059208  0.000000000
•   0.070  0.05  0.8180313  0.589728310
•   0.070  0.10  0.8584971  0.688028701
•   0.070  0.25  0.8871063  0.755059387
•   0.070  0.50  0.9080213  0.802613302
•   0.070  0.75  0.9166220  0.822096732
•   0.070  1.00  0.9205140  0.830835197
•   0.070  1.50  0.9224137  0.834881607
•   0.070  2.00  0.9219605  0.833912150
•   0.070  5.00  0.9236794  0.837693085
•   0.080  0.00        NaN          NaN
•   0.080  0.01  0.6059208  0.000000000
•   0.080  0.05  0.7956701  0.533813134
•   0.080  0.10  0.8476349  0.661878686
•   0.080  0.25  0.8782324  0.734330843
•   0.080  0.50  0.9036752  0.792783878
•   0.080  0.75  0.9107377  0.809034792
•   0.080  1.00  0.9154435  0.819612718
•   0.080  1.50  0.9190652  0.827360576
•   0.080  2.00  0.9202400  0.829895182
•   0.080  5.00  0.9213258  0.832510956
•   0.090  0.00        NaN          NaN
•   0.090  0.01  0.6059208  0.000000000
•   0.090  0.05  0.7737631  0.477748297
•   0.090  0.10  0.8347784  0.630774051
•   0.090  0.25  0.8722578  0.720097471
•   0.090  0.50  0.8989678  0.782087004
•   0.090  0.75  0.9060296  0.798518160
•   0.090  1.00  0.9133619  0.814761390
•   0.090  1.50  0.9149913  0.818356296
```

```
•   0.090  2.00  0.9172531  0.823260687
•   0.090  5.00  0.9181579  0.825492382
•   0.100  0.00        NaN          NaN
•   0.100  0.01  0.6059208  0.000000000
•   0.100  0.05  0.7557468  0.429986400
•   0.100  0.10  0.8221046  0.599631488
•   0.100  0.25  0.8641115  0.701029612
•   0.100  0.50  0.8949840  0.772741083
•   0.100  0.75  0.9024972  0.790366549
•   0.100  1.00  0.9088348  0.804647463
•   0.100  1.50  0.9132708  0.814427883
•   0.100  2.00  0.9140850  0.816211430
•   0.100  5.00  0.9148993  0.818373209
•   0.250  0.00        NaN          NaN
•   0.250  0.01  0.6059208  0.000000000
•   0.250  0.05  0.6091805  0.010005593
•   0.250  0.10  0.6606032  0.163831191
•   0.250  0.25  0.7655252  0.454299495
•   0.250  0.50  0.8293518  0.615865674
•   0.250  0.75  0.8504455  0.667206763
•   0.250  1.00  0.8588643  0.687805341
•   0.250  1.50  0.8648403  0.702232985
•   0.250  2.00  0.8636633  0.699715695
•   0.250  5.00  0.8634814  0.699964455
•   0.500  0.00        NaN          NaN
•   0.500  0.01  0.6059208  0.000000000
•   0.500  0.05  0.6059208  0.000000000
•   0.500  0.10  0.6234844  0.053428249
•   0.500  0.25  0.6812421  0.222603776
•   0.500  0.50  0.7538483  0.421540197
•   0.500  0.75  0.8044558  0.553049760
•   0.500  1.00  0.8148669  0.579436991
•   0.500  1.50  0.8226538  0.599452026
•   0.500  2.00  0.8221098  0.598402781
•   0.500  5.00  0.8222917  0.599050174
•   0.750  0.00        NaN          NaN
•   0.750  0.01  0.6059208  0.000000000
•   0.750  0.05  0.6059208  0.000000000
•   0.750  0.10  0.6199545  0.042790471
•   0.750  0.25  0.6669399  0.181487919
•   0.750  0.50  0.7316678  0.362403367
•   0.750  0.75  0.7918719  0.520235922
•   0.750  1.00  0.7988433  0.538198789
•   0.750  1.50  0.8032798  0.550233832
•   0.750  2.00  0.8031894  0.550195867
•   0.750  5.00  0.8028271  0.549603309
•   0.900  0.00        NaN          NaN
•   0.900  0.01  0.6059208  0.000000000
•   0.900  0.05  0.6059208  0.000000000
```

```
•   0.900  0.10  0.6168753  0.033460547
•   0.900  0.25  0.6614162  0.165530625
•   0.900  0.50  0.7263268  0.347764690
•   0.900  0.75  0.7855347  0.504115392
•   0.900  1.00  0.7949506  0.528204315
•   0.900  1.50  0.7982999  0.537309412
•   0.900  2.00  0.7988431  0.538673148
•   0.900  5.00  0.7975755  0.536186358
•
• Accuracy was used to select the optimal model using the largest value.
• The final values used for the model were sigma = 0.01 and C = 5.
```




```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 531  38
         1  26 324

               Accuracy : 0.9304
                 95% CI : (0.9119, 0.946)
    No Information Rate : 0.6061
```

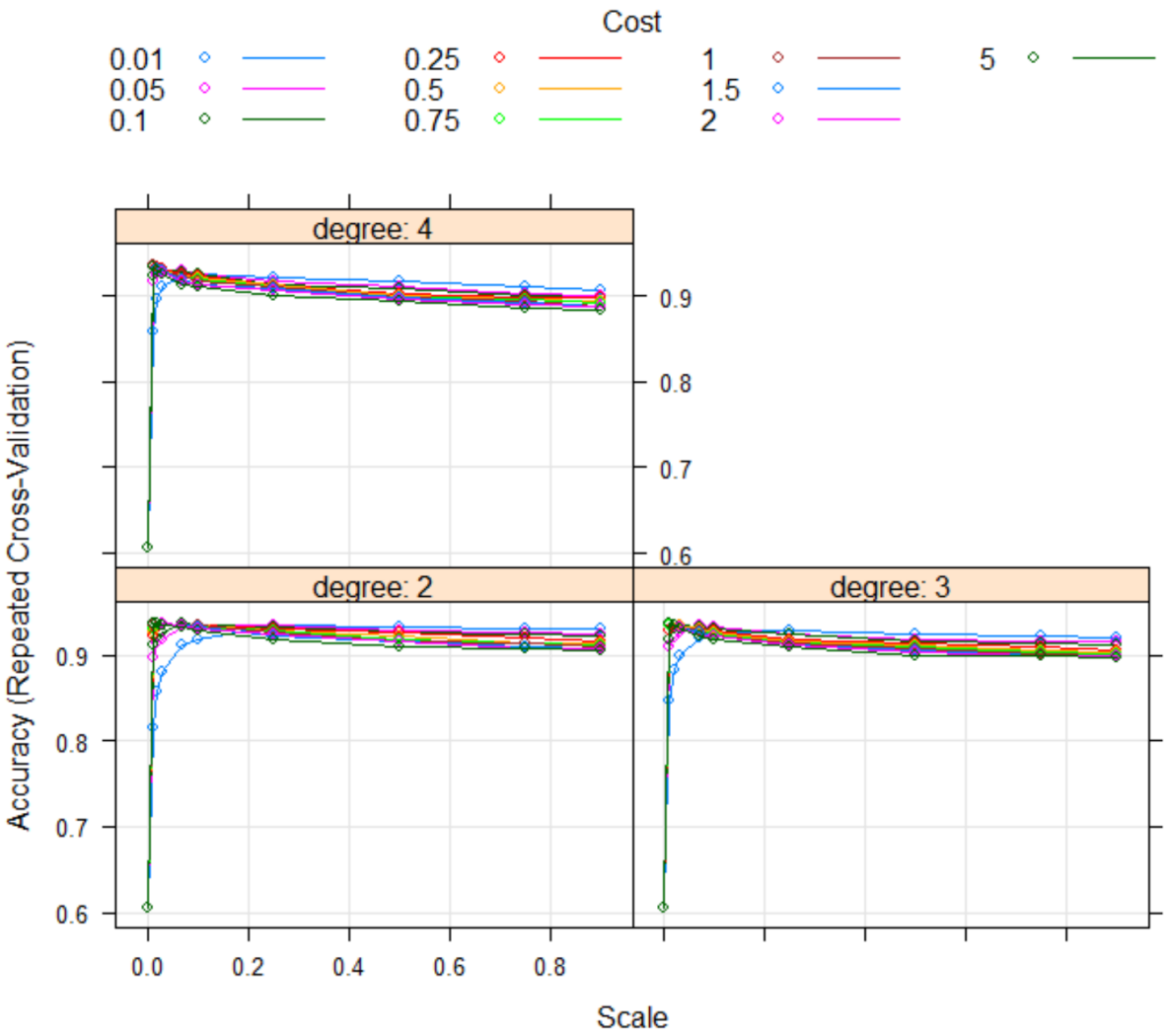
```
   P-Value [Acc > NIR] : <2e-16

                 Kappa : 0.8533
 Mcnemar's Test P-Value : 0.1691

            Sensitivity : 0.9533
            Specificity : 0.8950
         Pos Pred Value : 0.9332
         Neg Pred Value : 0.9257
             Prevalence : 0.6061
         Detection Rate : 0.5778
   Detection Prevalence : 0.6192
      Balanced Accuracy : 0.9242

       'Positive' Class : 0
```

3. Polynomial SVM Model:

- Use grid to find optimal Cost C parameter, degree and scale parameters.
- Must retune and adjust values repeatedly.

Training Model:

- Best degree: 2
- Best C: 5
- Best scale: 01



Test Set Results:

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 529 39
         1 28 323

               Accuracy: 0.9271
                 95% CI: (0.9083, 0.9431)
    No Information Rate: 0.6061
    P-Value [Acc > NIR]: <2e-16

                  Kappa: 0.8465
 Mcnemar's Test P-Value: 0.2218

            Sensitivity: 0.9497
            Specificity: 0.8923
         Pos Pred Value: 0.9313
         Neg Pred Value: 0.9202
```

```
            Prevalence: 0.6061
        Detection Rate: 0.5756
   Detection Prevalence: 0.6181
      Balanced Accuracy: 0.9210

       'Positive' Class : 0
```

Bayesian Methods:

Bayesglm:

Test Set Results:

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 531  38
         1  26 324

               Accuracy : 0.9304
                 95% CI : (0.9119, 0.946)
    No Information Rate : 0.6061
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8533
 Mcnemar's Test P-Value : 0.1691

            Sensitivity : 0.9533
            Specificity : 0.8950
         Pos Pred Value : 0.9332
         Neg Pred Value : 0.9257
             Prevalence : 0.6061
         Detection Rate : 0.5778
   Detection Prevalence : 0.6192
      Balanced Accuracy : 0.9242

       'Positive' Class : 0
```