Predicting Auto Insurance Claims

Kazeem Kareem, Alan Bouwman
Michigan Tech University
December 2022

Michigan Technological University

Our Goal

- Build models that would effectively predict whether or not an insurance policyholder flags an auto claim.
- Determine which model best suits this type of data by training several models.
- Determine what the most important predictors are for predicting insurance claims.

Our Dataset

Our dataset includes the following

- 10296 sample
- 10 categorical predictors
- 14 Numerical predictors
- **Unbalanced binary response variable** (Clm_flag: 73% No, 27% Yes)

Categorical Predictors - 10 Predictors

Predictor	Description	Levels
CAR_USE	Use of the car	Commercial, private
CAR_TYPE	Body type of the car	Panel Truck, Pickup, Sedan, Sports Car, SUV, Van
RED_CAR	If the car is red or not	Yes, No
REVOLKED	If driver's license has been revoked in the last 5 years	Yes, No
GENDER	Gender of driver - illegal to rate on in MI	F, M
MARRIED	Married or not	Yes, No
PARENT1	Single Parent	Yes, No
JOBCLASS	Job class	Unknown, BlueColar, Clerical, Doctor, Homemaker, Lawyer, Manager, Professional, Student
MAX_EDUC	Maximum education of driver - illegal to rate on in MI	<HighSchool, HighSchool, Bachelors, Masters, PhD
AREA	Area of work/home	Rural, Urban

Numerical Predictors - 14 Predictors

CLM_FREQ5

CLM_AMT5

KIDSDRIV

TRAVTIME

BLUEBOOK

RETAINED

NPOLICY

MVR_PTS

AGE

HOMEKIDS

YOJ - includes NaNs

INCOME - includes NaNs

HOME_VAL - includes NaNs

SAMEHOME - includes NaNs

Preprocessing

- **Imputation** — KNN imputation with K=5.

Dummy Variables

- Number of predictors before: 20
- Original categoricals, and first dummies dropped
- Number of predictors after: 37

No highly correlated predictors

No Near-Zero-Variance predictors

Transformations

- Center and scale
- Boxcox for (skewed numerical predictors)
- Spatial Sign (for outliers)

Data Spending

- 80/20 Train/Test split
- 10-fold Cross Validation
- Stratified random sampling for all splits

Considered Models

Linear

- Logistic Regression
- Penalized Logistic Regression
- Partial Least Squares
- Linear Discriminant Analysis

Non-Linear

- Neural Network
- SVM
- KNN
- Random GLM Ensemble
- Random Forest
- Naive Bayes
- Quadratic Discriminant Analysis

Performance Metric: Kappa

Logistic Regression

```
> logistic.glm
Generalized Linear Model

8237 samples
  37 predictor
   2 classes: 'No', 'Yes'

Pre-processing: Box-Cox transformation (14), centered (37), scaled (37), spatial
 sign transformation (37)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 7413, 7413, 7413, 7414, 7413, 7412, ...
Resampling results:

  Accuracy   Kappa
  0.7858456  0.3885542
```

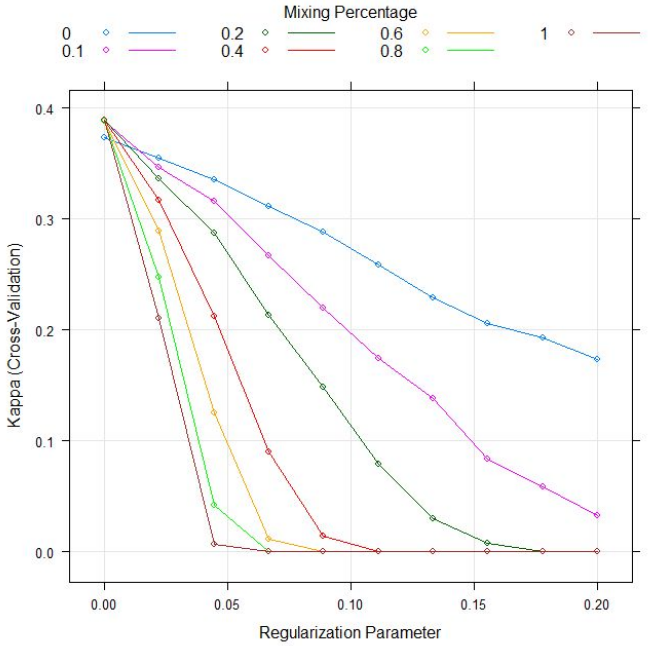
Prediction on training set

Kappa = 0.3922

Accuracy = 0.7874

Penalized Logistic Regression

```
> glmnTuned$bestTune
   alpha lambda
21   0.2      0
> res[c("alpha", "lambda", disp_metric)]
   alpha     lambda       Kappa  Accuracy
1    0.0 0.00000000 0.373085921 0.7843805
2    0.0 0.02222222 0.355113597 0.7824385
3    0.0 0.04444444 0.335418329 0.7817923
4    0.0 0.06666667 0.311087674 0.7790741
5    0.0 0.08888889 0.288052523 0.7758363
6    0.0 0.11111111 0.258238284 0.7710444
7    0.0 0.13333333 0.228875105 0.7662535
8    0.0 0.15555556 0.206111313 0.7631457
9    0.0 0.17777778 0.193288073 0.7618502
10   0.0 0.20000000 0.172972984 0.7586130
11   0.1 0.00000000 0.388337932 0.7860637
12   0.1 0.02222222 0.346814781 0.7816616
13   0.1 0.04444444 0.315985175 0.7802387
14   0.1 0.06666667 0.267185962 0.7729865
15   0.1 0.08888889 0.219522310 0.7659946
16   0.1 0.11111111 0.174199239 0.7600377
17   0.1 0.13333333 0.138249837 0.7548579
18   0.1 0.15555556 0.083902444 0.7461809
19   0.1 0.17777778 0.058426732 0.7429429
20   0.1 0.20000000 0.032447351 0.7391866
21   0.2 0.00000000 0.388798212 0.7861931
22   0.2 0.02222222 0.336650152 0.7803674
23   0.2 0.04444444 0.286921031 0.7758351
24   0.2 0.06666667 0.213436276 0.7652170
25   0.2 0.08888889 0.148836905 0.7562824
```



Tuning: Lambda between 0 and 0.2, length 10. Alpha between 0 and 1.

Kappa=0.388

Accuracy=0.786

Alpha=0.2, Lambda=0.

As the penalty increases, the model tends to predict "No" more often

Michigan Technological University

Linear Discriminant Analysis

```
> lda
Linear Discriminant Analysis

8237 samples
  37 predictor
   2 classes: 'No', 'Yes'

Pre-processing: centered (37), scaled (37)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 7413, 7413, 7413, 7414, 7413, 7412, ...
Resampling results:

  Accuracy   Kappa
  0.7870601  0.3869664
```

Prediction on training set

Kappa = 0.391

Accuracy = 0.789

Quadratic Discriminant Analysis

```
Quadratic Discriminant Analysis

8237 samples
  37 predictor
   2 classes: 'No', 'Yes'

Pre-processing: centered (37), scaled (37)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 7413, 7413, 7413, 7414, 7413, 7412, ...
Resampling results:

  Accuracy   Kappa
  0.7451786  0.4041244
```

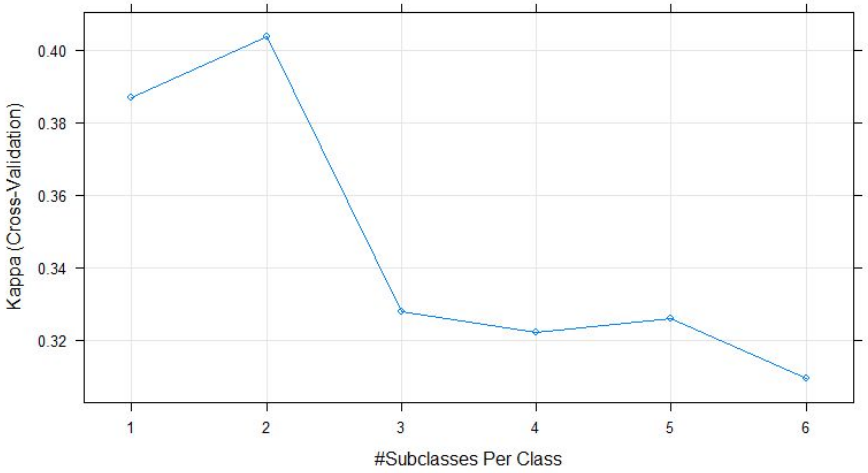
Prediction on training set

Kappa = 0.435

Accuracy = 0.758

Mixture Discriminant Analysis

```
Mixture Discriminant Analysis

8237 samples
  37 predictor
   2 classes: 'No', 'Yes'

Pre-processing: centered (37), scaled (37)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 7413, 7413, 7413, 7414, 7413, 7412, ...
Resampling results across tuning parameters:

  subclasses  Accuracy   Kappa
  1           0.7870601  0.3869664
  2           0.7849987  0.4038203
  3           0.7601012  0.3277264
  4           0.7587677  0.3221711
  5           0.7614681  0.3258798
  6           0.7578042  0.3093804

Kappa was used to select the optimal model using the largest value.
The final value used for the model was subclasses = 2.
```



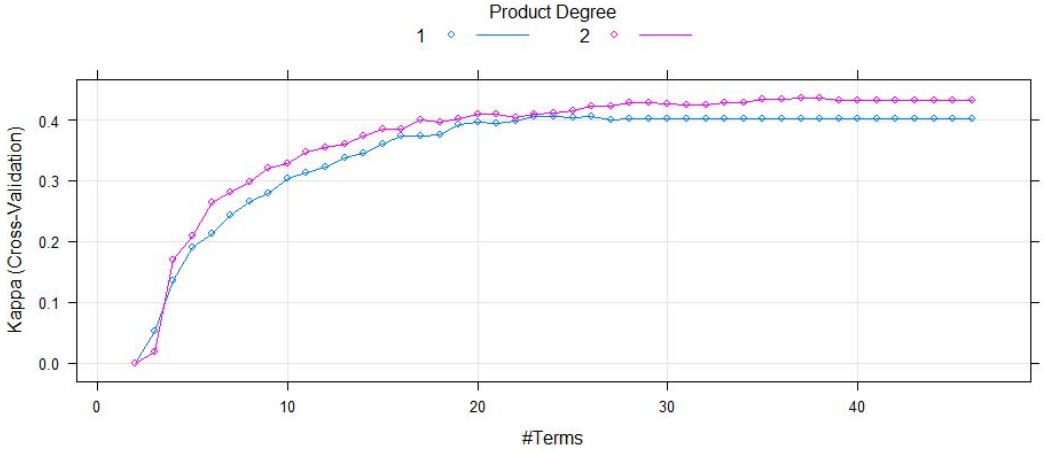
Prediction on training set

Kappa: 0.436

Accuracy: 0.794

Flexible Discriminant Analysis

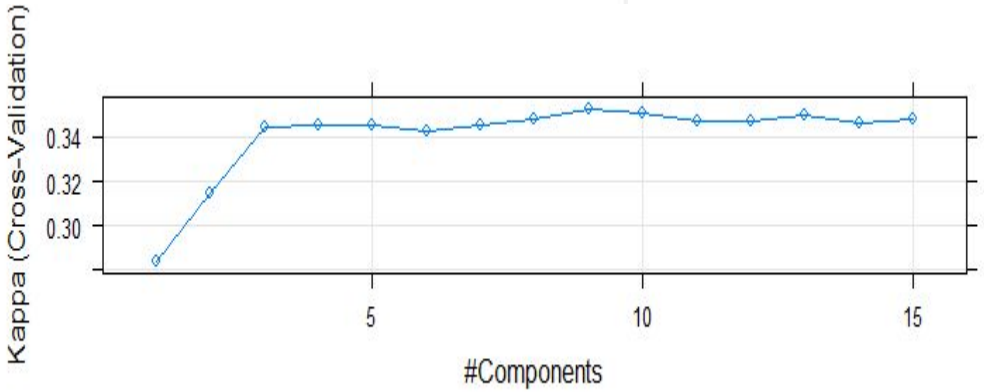
```
Flexible Discriminant Analysis

8237 samples
  37 predictor
   2 classes: 'No', 'Yes'

Pre-processing: centered (37), scaled (37)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 7413, 7413, 7413, 7414, 7413, 7412, ...
Resampling results across tuning parameters:

  degree  nprune  Accuracy   Kappa
  1        2      0.7338838  0.00000000
  1        3      0.7340062  0.05246859
  1        4      0.7336402  0.13588997
  1        5      0.7439641  0.19105365
  1        6      0.7500343  0.21310526
  1        7      0.7541611  0.24362922
  1        8      0.7591377  0.26501456
  1        9      0.7624165  0.27901384
  1       10      0.7677579  0.30423131
  1       11      0.7698212  0.31270668
  1       12      0.7722486  0.32230605
  1       13      0.7751648  0.33735171
  1       14      0.7784394  0.34528270
  1       15      0.7818412  0.35992975
  1       16      0.7860874  0.37373969
  1       17      0.7860879  0.37363169
  1       18      0.7868166  0.37613832
  1       19      0.7911880  0.39267148
  1       20      0.7919156  0.39691018
  1       21      0.7913070  0.39453618
  1       22      0.7917915  0.39897989
  1       23      0.7938554  0.40587077
  1       24      0.7942185  0.40626101
  1       25      0.7933702  0.40434663
  1       26      0.7933702  0.40598578
  1       27      0.7916713  0.40066495
  1       28      0.7919146  0.40197235
  1       29      0.7922781  0.40267708
  1       30      0.7919136  0.40115845
  1       31      0.7920351  0.40140565
  1       32      0.7920351  0.40140565
  1       33      0.7920351  0.40140565
  1       34      0.7920351  0.40140565
  1       35      0.7920351  0.40140565
  1       36      0.7920351  0.40140565
  1       37      0.7920351  0.40140565
  1       38      0.7920351  0.40140565
  2        2      0.7338838  0.00000000
  2        3      0.7342480  0.01815657
  2        4      0.7423785  0.17020573
  2        5      0.7459048  0.20873291
  2        6      0.7542823  0.26366082
  2        7      0.7590174  0.28176964
  2        8      0.7616898  0.29752280
  2        9      0.7695808  0.32014818
  2       10      0.7712795  0.32913746
  2       11      0.7756519  0.34727546
  2       12      0.7772285  0.35389511
  2       13      0.7797791  0.36102558
  2       14      0.7835445  0.37437083
  2       15      0.7862142  0.38434183
  2       16      0.7859687  0.38459706
  2       17      0.7903394  0.39949497
  2       18      0.7897328  0.39695172
  2       19      0.7907031  0.40111243
  2       20      0.7933734  0.40986451
  2       21      0.7927666  0.40994717
  2       22      0.7900952  0.40344968
  2       23      0.7920386  0.40958238
  2       24      0.7928881  0.41145398
  2       25      0.7942234  0.41527923
  2       26      0.7956801  0.42267652
  2       27      0.7958010  0.42303183
  2       28      0.7970167  0.42813696
  2       29      0.7968944  0.42880730
  2       30      0.7962865  0.42689719
  2       31      0.7953156  0.42405455
  2       32      0.7955586  0.42532592
  2       33      0.7967731  0.42928628
  2       34      0.7970155  0.42922560
  2       35      0.7983519  0.43435860
  2       36      0.7984724  0.43440464
  2       37      0.7989575  0.43579194
  2       38      0.7988366  0.43537284

Kappa was used to select the optimal model using the largest value.
The final values used for the model were degree = 2 and nprune = 37.
```



PLSDA

```
Partial Least Squares

8237 samples
  37 predictor
   2 classes: 'No', 'Yes'

Pre-processing: centered (37), scaled (37), Box-Cox transformation (14), spatial sign
 transformation (37)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 7414, 7413, 7413, 7412, 7413, 7414, ...
Resampling results across tuning parameters:

  ncomp  Accuracy   Kappa
   1     0.7669090  0.2836473
   2     0.7732208  0.3148553
   3     0.7818378  0.3448304
   4     0.7817178  0.3457793
   5     0.7815965  0.3453488
   6     0.7812319  0.3427269
   7     0.7817172  0.3452732
   8     0.7824459  0.3478793
   9     0.7841461  0.3530215
  10     0.7835390  0.3508100
  11     0.7820824  0.3473202
  12     0.7819608  0.3469116
  13     0.7828113  0.3502085
  14     0.7815964  0.3461161
  15     0.7822038  0.3478662

Kappa was used to select the optimal model using the largest value.
The final value used for the model was ncomp = 9.
```

Kappa: 0.359
Accuracy: 0.786

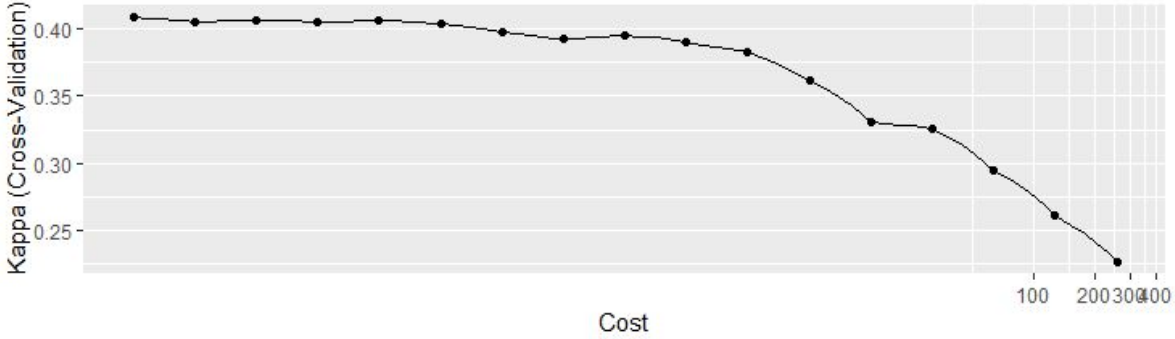


Michigan Technological University

SVM

```
Support Vector Machines with Radial Basis Function Kernel

8237 samples
  37 predictor
   2 classes: 'No', 'Yes'

Pre-processing: centered (37), scaled (37)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 7414, 7413, 7413, 7414, 7414, 7412, ...
Resampling results across tuning parameters:

  C            Accuracy   Kappa
  3.90625e-03  0.7808662  0.4086761
  7.81250e-03  0.7797734  0.4053205
  1.56250e-02  0.7805025  0.4068389
  3.12500e-02  0.7795308  0.4051699
  6.25000e-02  0.7796517  0.4062849
  1.25000e-01  0.7839006  0.4036782
  2.50000e-01  0.7900921  0.3975365
  5.00000e-01  0.7920346  0.3921301
  1.00000e+00  0.7942217  0.3956954
  2.00000e+00  0.7920974  0.3898480
  4.00000e+00  0.7905635  0.3822644
  8.00000e+00  0.7860188  0.3614540
  1.60000e+01  0.7757184  0.3308822
  3.20000e+01  0.7790555  0.3259054
  6.40000e+01  0.7722017  0.2947362
  1.28000e+02  0.7652715  0.2613226
  2.56000e+02  0.7590569  0.2271664

Tuning parameter 'sigma' was held constant at a value of 0.009752278
Kappa was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.009752278 and C = 0.00390625.
```

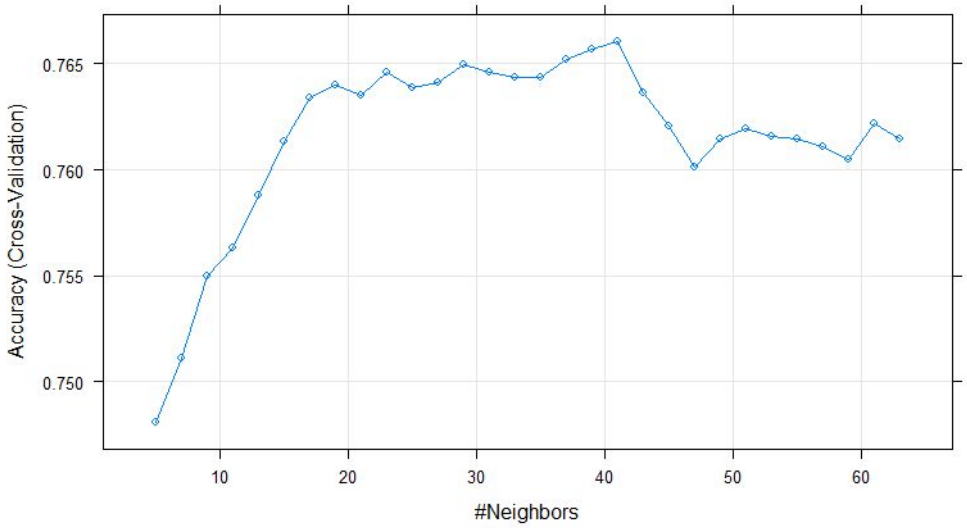
Kappa: 0.4348760
Accuracy: 0.7903363



KNN

```
k-Nearest Neighbors

8237 samples
  37 predictor
   2 classes: 'No', 'Yes'

Pre-processing: centered (37), scaled (37)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 7413, 7413, 7413, 7414, 7413, 7412, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   5  0.7480903  0.2810184
   7  0.7511270  0.2668240
   9  0.7550113  0.2657393
  11  0.7563454  0.2630208
  13  0.7587727  0.2604617
  15  0.7613216  0.2603764
  17  0.7633840  0.2639692
  19  0.7639925  0.2577572
  21  0.7635080  0.2541458
  23  0.7645986  0.2525687
  25  0.7638717  0.2453525
  27  0.7641140  0.2413780
  29  0.7649644  0.2389830
  31  0.7646006  0.2360319
  33  0.7643576  0.2333843
  35  0.7643587  0.2300928
  37  0.7652102  0.2310049
  39  0.7656952  0.2293877
  41  0.7660577  0.2278178
  43  0.7636301  0.2186441
  45  0.7620524  0.2121380
  47  0.7601090  0.2030595
  49  0.7614449  0.2070724
  51  0.7619306  0.2070819
  53  0.7615661  0.2025984
  55  0.7614456  0.2004377
  57  0.7610809  0.1971856
  59  0.7604741  0.1946524
  61  0.7621729  0.1981952
  63  0.7614441  0.1951924

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 41.
```

Kappa: 0.2542816
Accuracy: 0.7722472



Naive Bayes

No Tuning Parameters
Did not perform very well

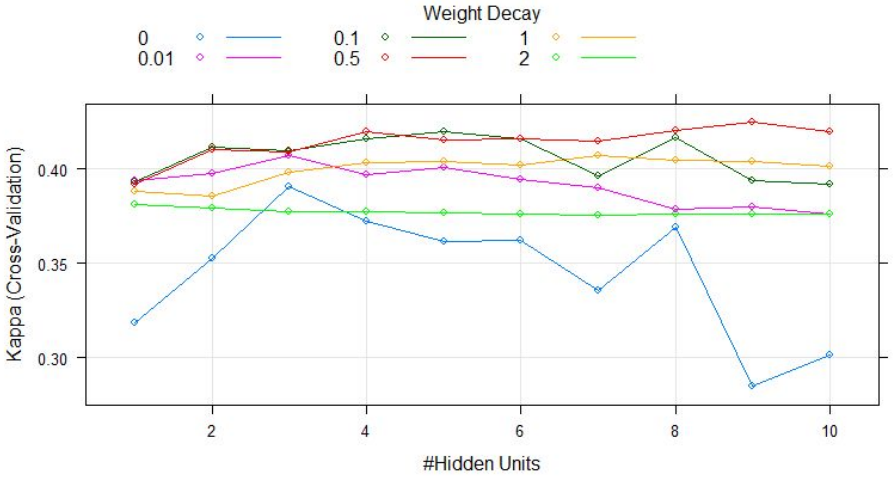
```
Naive Bayes

7722 samples
  37 predictor
   2 classes: 'No', 'Yes'

Pre-processing: centered (37), scaled (37), Box-Cox transformation (5), spatial sign transformation (37)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 6951, 6949, 6950, 6950, 6950, 6949, ...
Resampling results:

  logLoss   AUC       prAUC      Accuracy   Kappa      F1         Sensitivity  Specificity  Pos_Pred_Value  Neg_Pred_Value  Precision  Recall     Detection_Rate
  1.726712  0.709419  0.6543539  0.7068175  0.2314837  0.8027729  0.8136622    0.4121217    0.7923987       0.4463602       0.7923987  0.8136622  0.5971351
  Balanced_Accuracy
  0.612892

Tuning parameter 'laplace' was held constant at a value of 1
Tuning parameter 'usekernel' was held constant at a value of TRUE
Tuning parameter 'adjust' was
 held constant at a value of 1
  logLoss   AUC       prAUC      Accuracy   Kappa      F1         Sensitivity  Specificity  Pos_Pred_Value  Neg_Pred_Value  Precision  Recall     Detection_Rate
  1.726712  0.709419  0.6543539  0.7068175  0.2314837  0.8027729  0.8136622    0.4121217    0.7923987       0.4463602       0.7923987  0.8136622  0.5971351
  Balanced_Accuracy
  0.612892

Tuning parameter 'laplace' was held constant at a value of 1
Tuning parameter 'usekernel' was held constant at a value of TRUE
Tuning parameter 'adjust' was
 held constant at a value of 1
>
```

Kappa = 0.270
Accuracy = 0.718
(This is worse than always guessing “No”)

Neural Network

```
Neural Network

8237 samples
  37 predictor
   2 classes: 'No', 'Yes'

Pre-processing: centered (37), scaled (37), Box-Cox transformation (14), spatial
 sign transformation (37)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 7413, 7413, 7413, 7414, 7413, 7412, ...
Resampling results across tuning parameters:

  size  decay  Accuracy   Kappa
   1    0.00   0.7767436  0.3183810
   1    0.01   0.7862100  0.3937701
   1    0.10   0.7856029  0.3934737
   1    0.50   0.7856029  0.3920128
   1    1.00   0.7857241  0.3882162
   1    2.00   0.7863316  0.3812461
   2    0.00   0.7803859  0.3528780
   2    0.01   0.7876704  0.3976537
   2    0.10   0.7900961  0.4114992
   2    0.50   0.7909446  0.4104459
   2    1.00   0.7852397  0.3856608
   2    2.00   0.7864533  0.3790724
   3    0.00   0.7811145  0.3905262
   3    0.01   0.7865745  0.4073332
   3    0.10   0.7885173  0.4094857
   3    0.50   0.7913094  0.4091216
   3    1.00   0.7892467  0.3980562
   3    2.00   0.7862107  0.3775693
   4    0.00   0.7693363  0.3720631
   4    0.01   0.7829331  0.3966769
   4    0.10   0.7902173  0.4158427
   4    0.50   0.7948315  0.4196700
   4    1.00   0.7911894  0.4029843
   4    2.00   0.7862108  0.3771189
   5    0.00   0.7689744  0.3613851
   5    0.01   0.7832967  0.4010319
   5    0.10   0.7905805  0.4196194
   5    0.50   0.7931314  0.4153444
   5    1.00   0.7915542  0.4042231
   5    2.00   0.7862108  0.3765101
   6    0.00   0.7675174  0.3619748
   6    0.01   0.7805039  0.3945880
   6    0.10   0.7891277  0.4159793
   6    0.50   0.7922812  0.4160779
   6    1.00   0.7909472  0.4019617
   6    2.00   0.7860895  0.3760433
   7    0.00   0.7633898  0.3355136
   7    0.01   0.7767383  0.3899792
   7    0.10   0.7806230  0.3962571
   7    0.50   0.7917957  0.4146275
   7    1.00   0.7927687  0.4073002
   7    2.00   0.7859680  0.3755816
   8    0.00   0.7645937  0.3688561
   8    0.01   0.7722467  0.3785666
   8    0.10   0.7881529  0.4167029
   8    0.50   0.7941021  0.4200931
   8    1.00   0.7919187  0.4043467
   8    2.00   0.7860892  0.3758282
   9    0.00   0.7548845  0.2847842
   9    0.01   0.7715222  0.3799756
   9    0.10   0.7784406  0.3940739
   9    0.50   0.7956810  0.4245841
   9    1.00   0.7919174  0.4039382
   9    2.00   0.7862104  0.3758552
  10    0.00   0.7545262  0.3011838
  10    0.01   0.7706708  0.3763468
  10    0.10   0.7789265  0.3921353
  10    0.50   0.7932531  0.4196246
  10    1.00   0.7911892  0.4013713
  10    2.00   0.7862104  0.3758552

Accuracy was used to select the optimal model using the largest value.
The final values used for the model were size = 9 and decay = 0.5.
```

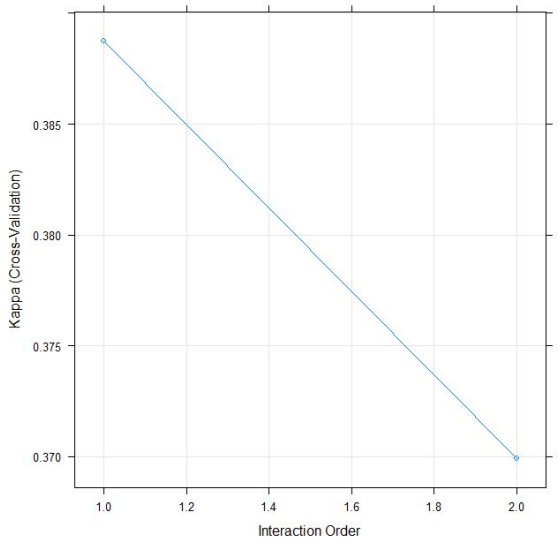
Kappa: 0.466
Accuracy: 0.809



Random GLM Ensemble

```
Pre-processing: centered (37), scaled (37), Box-Cox transformation (5), spatial sign transformation (37)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 6950, 6950, 6949, 6950, 6951, 6949, ...
Resampling results across tuning parameters:

  maxInteractionOrder  logLoss    AUC        prAUC      Accuracy   Kappa      F1         Sensitivity  Specificity  Pos_Pred_Value  Neg_Pred_Value  Precision
  1                    0.4514440  0.8113927  0.7563389  0.7872344  0.3887311  0.8634669  0.9167156    0.4301776    0.8161650       0.6520949       0.816165
  2                    0.4550463  0.8070595  0.7555861  0.7843820  0.3699282  0.8627455  0.9232415    0.4014516    0.8097618       0.6541849       0.809761
  Recall     Detection_Rate  Balanced_Accuracy
  0.9167156  0.6727548       0.6734466
  0.9232415  0.6775436       0.6623465
```

MaxInteractionOrder: 1
Training Time: Over 4 hours



Summary of Model Metrics

Model	Accuracy	Kappa	Sensitivity	Specificity	Precision	F1	AUC
logistic	0.787	0.392	0.915	0.437	0.817	0.863	0.812
penalized	0.789	0.398	0.916	0.440	0.819	0.864	0.816
LDA	0.789	0.391	0.919	0.428	0.816	0.865	0.808
QDA	0.758	0.435	0.781	0.696	0.876	0.826	0.795
MDA	0.794	0.436	0.896	0.512	0.835	0.865	0.747
PLSDA	0.786	0.359	0.936	0.370	0.804	0.865	0.804
random glm	0.789	0.392	0.919	0.429	0.816	0.865	0.816
NaiveBayes	0.716	0.256	0.819	0.431	0.799	0.809	0.737
KNN	0.758	0.309	0.893	0.385	0.800	0.844	0.778
SVM	0.789	0.435	0.883	0.532	0.839	0.860	0.797
FDA	0.809	0.462	0.920	0.501	0.836	0.876	0.800
NNetwork	0.809	0.466	0.918	0.508	0.837	0.876	0.804

Table 1: Performance Profiles of various Classification models

Best Model

Our best two models are

1. Neural Networks

   Train: (Kappa: 0.466; computation time: 19.3 minutes)

   Test: Kappa = 0.4165

2. Flexible Discriminant Analysis

   Train: (Kappa: 0.462; computation time: 26.53 minutes)

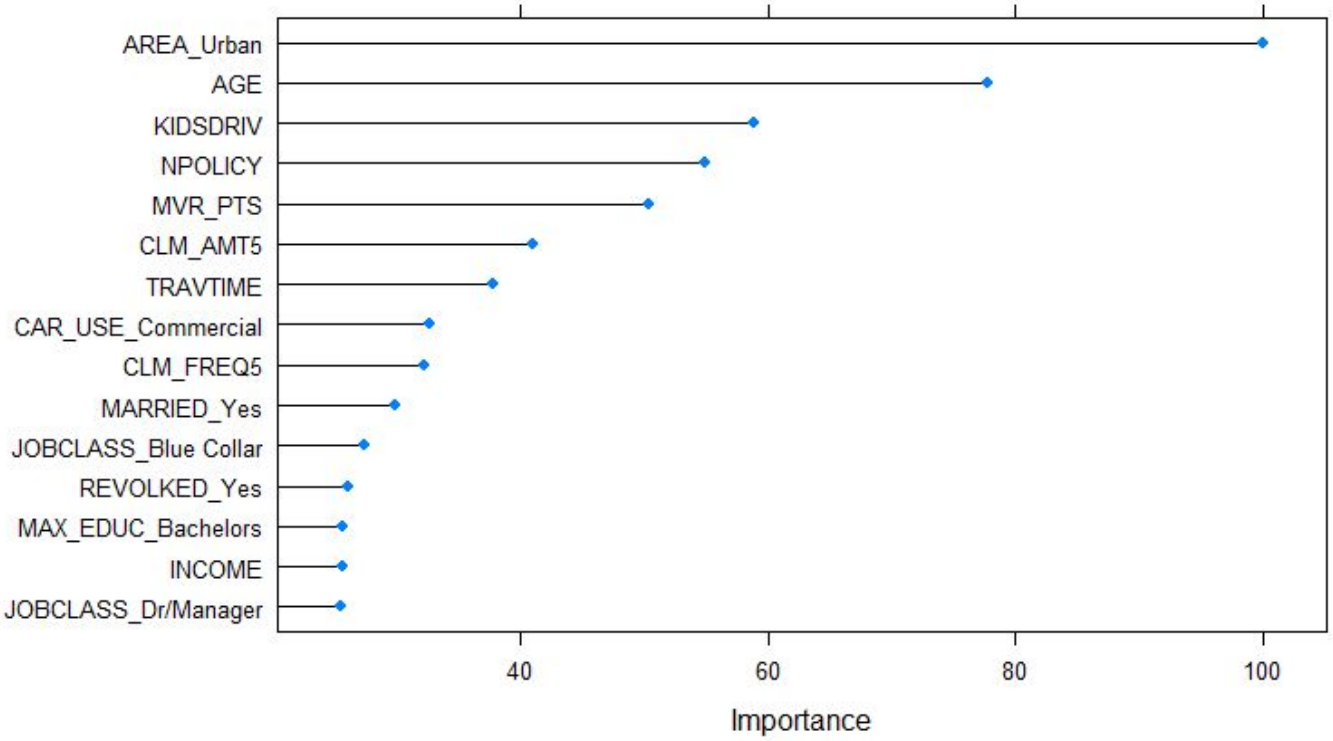
   Test: Kappa = 0.4138

Predicting with Best Model (Test set)

Neural Network

```
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  1375  291
       Yes  136  257

               Accuracy : 0.7926
                 95% CI : (0.7745, 0.8099
    No Information Rate : 0.7339
    P-Value [Acc > NIR] : 3.543e-10

                  Kappa : 0.4165

 Mcnemar's Test P-Value : 9.153e-14

            Sensitivity : 0.9100
            Specificity : 0.4690
         Pos Pred Value : 0.8253
         Neg Pred Value : 0.6539
             Prevalence : 0.7339
         Detection Rate : 0.6678
   Detection Prevalence : 0.8091
      Balanced Accuracy : 0.6895

       'Positive' Class : No
```

Important Variables



Will I get into an Accident?

```r
alan = data.frame("CLM_FREQ5"=0,    # How many accidents in the last 5 years
                  "CLM_AMT5"=0,     # Total cost of accidents in the last 5 years
                  "KIDSDRIV"=0,     # Number of kids who drive
                  "TRAVTIME"=10,    # Commute time to work
                  "BLUEBOOK"=4171,  # Current car price (can google)
                  "RETAINED"=8 ,    # How many years you have been with the insurance company on this policy
                  "NPOLICY"= 2,     # Number of policies
                  "MVR_PTS"=0,      # You can guess or buy this info online for $12 (plus some fees)
                  "AGE"=24,         # self explanatory
                  "HOMEKIDS"=0,     # number of kids that live with you
                  "YOJ"=4,          # years at your current job
                  "INCOME"=,        # annual income
                  "HOME_VAL"=0.0,   # value of your home (I rent so I put 0)
                  "SAMEHOME"=3,     # How many years you have lived in your house
                 ## The rest of the columns are for dummy vars and are either 0 or 1. Most are self explanatory
                 "CAR_USE_Commercial"=0, # the other option is private (so put zero unless you use your car commercially)
                 "CAR_TYPE_Pickup"=0,
                 "CAR_TYPE_Sedan"=1,
                 "CAR_TYPE_Sports Car"=0,
                 "CAR_TYPE_SUV"=0,
                 "CAR_TYPE_Van"=0,
                 "RED_CAR_yes"=1,
                 "REVOLKED_Yes"=0, # 1 if your driver's license has ever been revoked
                 "GENDER_M"=1,
                 "MARRIED_Yes"=0,
                 "PARENT1_Yes"=0, # 1 if you are a single parent
                 "JOBCLASS_Blue Collar"=0,
                 "JOBCLASS_Clerical"=0,
                 "JOBCLASS_Dr/Manager"=0,
                 "JOBCLASS_Home Maker"=0,
                 "JOBCLASS_Lawyer"=0,
                 "JOBCLASS_Professional"=0,
                 "JOBCLASS_Student"=1,
                 "MAX_EDUC_Bachelors"=1,
                 "MAX_EDUC_High School"=0,
                 "MAX_EDUC_Masters"=0,
                 "MAX_EDUC_PhD"=0,
                 "AREA_Urban"=0 )  # other option is rural. I think that's what houghton counts as
```

```
> predict(nnet, alan)
[1] No
Levels: No Yes
[1] No
```

Conclusion

- The best two models were Flexible Discriminant Analysis and Neural Network
- The most important variables were Area, Age and Number of kids who Drive
- The Discriminant Analysis Models performed very well
- Logistic model was also competitive and really fast to implement
- KNN and Naive Bayes performed the worst

Thank You for Listening!

Questions?