Model selection for Credit Card Approval

Shuyan Wan,

Hongfei Li,

Hao Hui

I. Introduction

Problem Description and background:

When a person applies for a new credit card, the credit card company would decide whether to issue him the card based on his personal information and financial record.

we would like to conduct a research related to credit card approval based on several factors which most banks consider.

Literature Review

- **three categories of major classification algorithms:**
- Decision Tree / Rule based Classifiers
- Statistical Classifiers;
- Neural Network Classifiers.

Decision Tree / Rule based Classifiers

- In a node m, representing a region Rm with Nm observations, let

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k),$$

the proportion of class k observations in node m. Then we classify the observations in node m to class k(m)= arg max k $\hat{p}_{mk}$, the majority class in node m.

II. Preliminary Analysis

- **Data Description**:

Our data set consists of 1319 applications for credit cards and their results (approved or rejected). The data comes from Professor William Greene's (New York University) on-line data for his book "Econometric Analysis, 5th Edition", provided by AE. (http://pages.stern.nyu.edu/~wgreene/Text/econometricanalysis.htm)

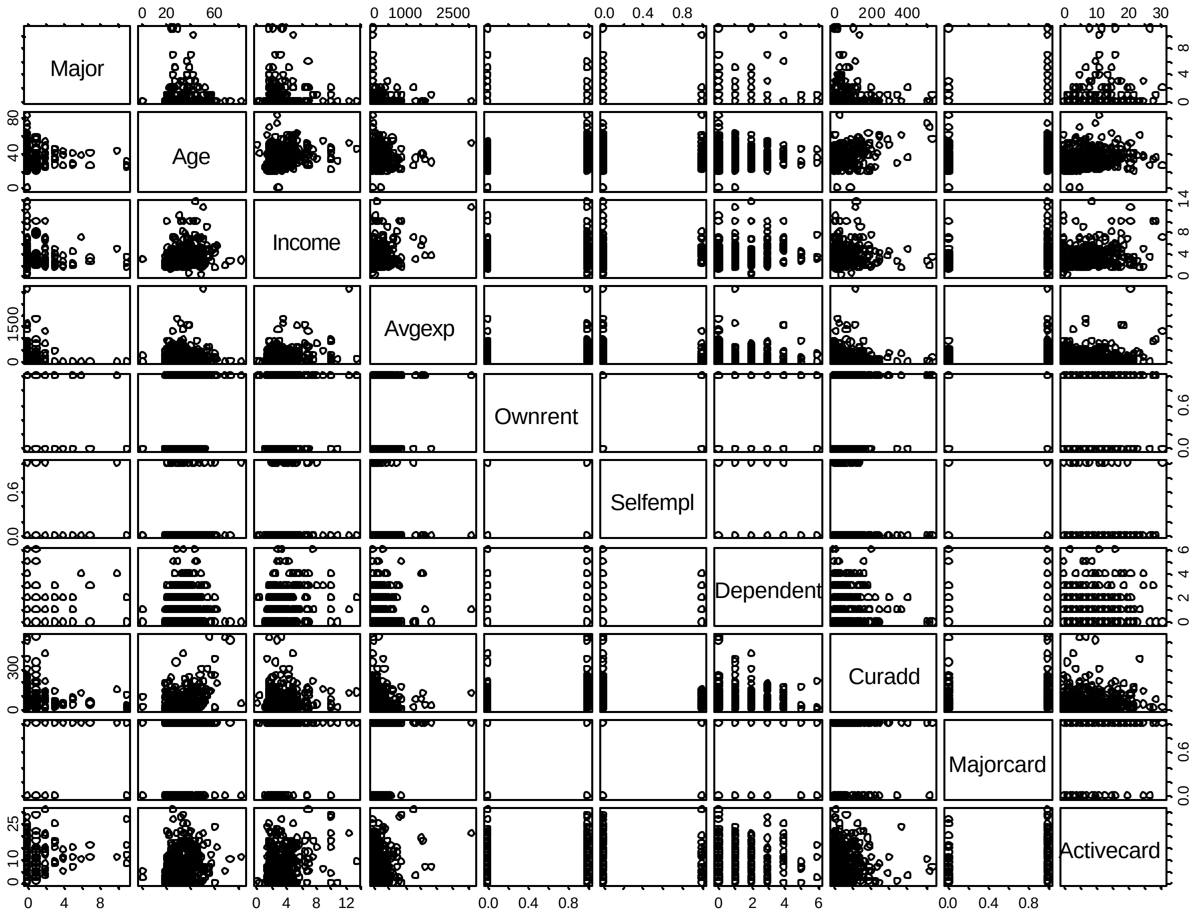
Two note-worthy features

- There are some missing data in the predictor “Age”. We will use nearest neighbor method to handle these.
- There are some people with same records but different results of their application, even their ages are the same. Use jittering for age.

Brief explanation of the variables:

- Approval = response/output. 1 if application for credit card accepted, 0 if not.
- Major = Number of major derogatory reports
- Age = Age n years plus twelfths of a year + jittering.
- Income = Yearly income (divided by 10,000)
- Avgexp = Average monthly credit card expenditure
- Ownrent = Dummy variable, 1 if owns his home, 0 if rent
- Selfempl = Dummy variable, 1 if self employed, 0 if not.
- Dependent = 1 + number of dependents, applicant himself is regarded as one dependent.
- Curadd = months living at current address
- ActiveCard = number of active credit accounts
- MajorCard = number of major credit cards held.

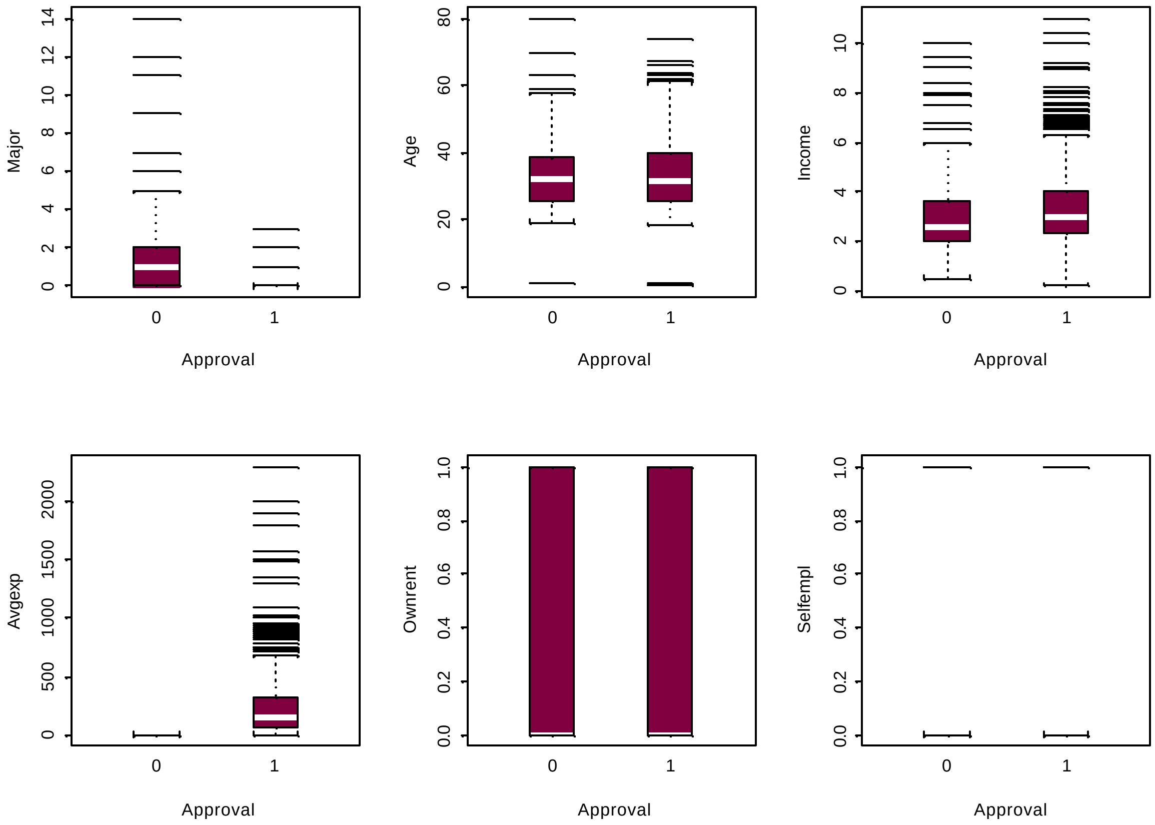
Graphical summaries:



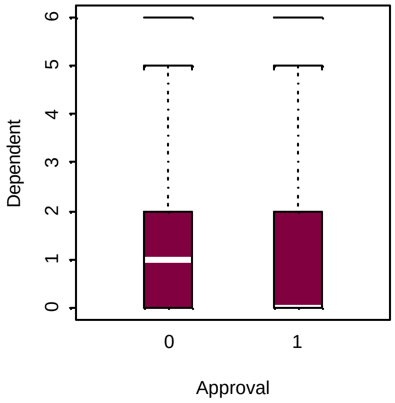
Correlation Matrix

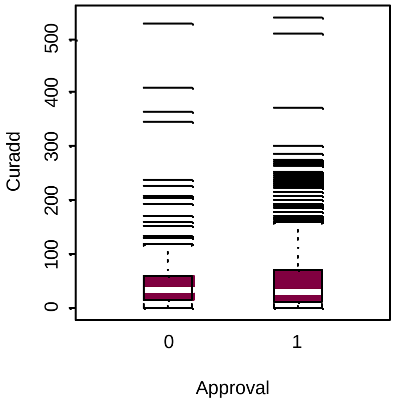
```
Correlation of Coefficients:
           (Intercept)  Major      Age        Income     Avgexp     Ownrent    Selfempl
     Major -0.0048614
       Age -0.6171099  -0.0424090
    Income -0.3005576   0.0405857 -0.1272267
    Avgexp -0.1018250  -0.0016668 -0.0118810 -0.0730608
   Ownrent  0.1501756   0.0361958 -0.2274511 -0.1517531  0.0508048
  Selfempl -0.0707626  -0.0799180  0.0569033 -0.2094046  0.1161601  0.0354740
 Dependent -0.1397503   0.0509208  0.0181660 -0.1111812 -0.0503892 -0.1152073 -0.0544166
    Curadd  0.1315405  -0.0566883 -0.4688740 -0.0007789 -0.0009823 -0.1922742 -0.0208237
 Majorcard -0.3925220  -0.0120166 -0.0747958 -0.1784211  0.0624550  0.0473558  0.0490531
Activecard -0.1201610  -0.2997068  0.0576801 -0.0748448 -0.1003962 -0.2173886  0.0799234
           Dependent  Curadd     Majorcard
     Major
       Age
    Income
    Avgexp
   Ownrent
  Selfempl
 Dependent
    Curadd  0.1221446
 Majorcard  0.0201242  0.1340061
Activecard -0.0226648  0.0034587 -0.0497721
```

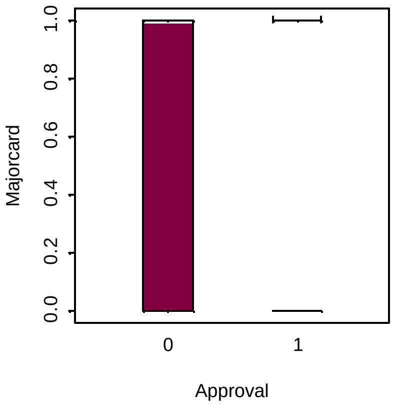
Boxplots

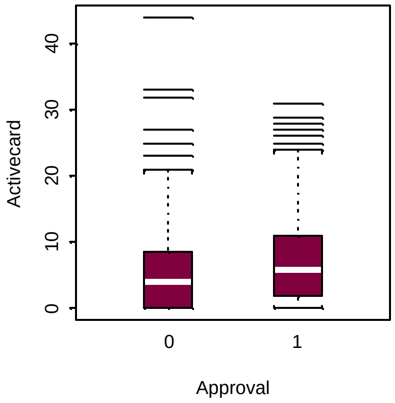

Major
0
2
4
6
8
10
12
14
0
1
Approval
Age
0
20
40
60
80
0
1
Approval
Income
0
2
4
6
8
10
0
1
Approval
Avgexp
0
500
1000
1500
2000
0
1
Approval
Ownrent
0.0
0.2
0.4
0.6
0.8
1.0
0
1
Approval
Selfempl
0.0
0.2
0.4
0.6
0.8
1.0
0
1
Approval

Boxplots(II) and Table









	Selfempl(0-no)	Selfempl(1-yes)
Approval(0-no)	268	28
Approval(1-yes)	960	63

III. Main Analysis

- **Method I. Classification tree**

summary for the training data:

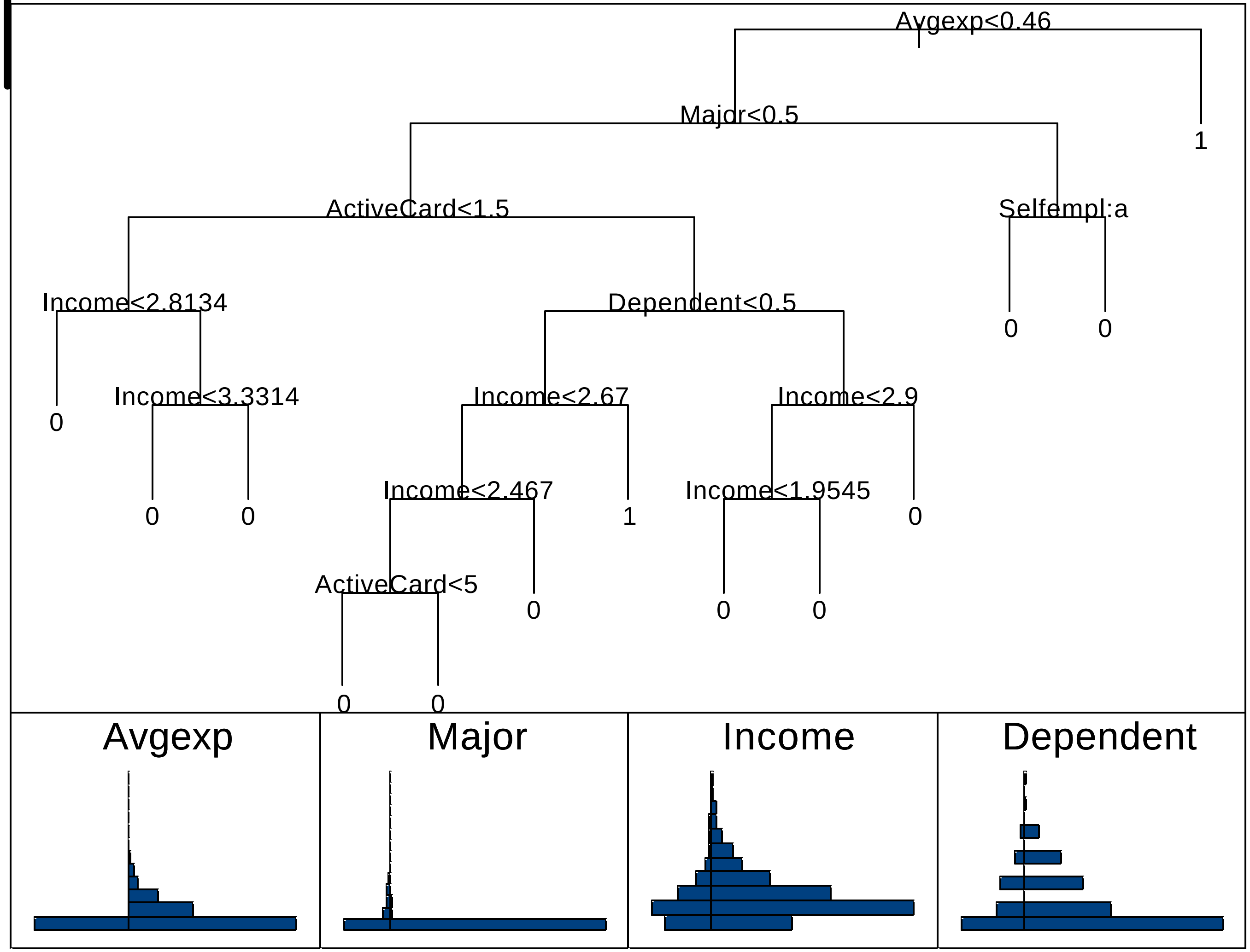
```
tree(formula = Approval ~ Selfempl + Ownrent + Majorcard + Major
    + Income +
    Avgexp + Dependent + Curadd + ActiveCard, data = card)
Variables actually used in tree construction:
[1] "Avgexp"     "Major"      "ActiveCard" "Income"     "Dependent"
[6] "Selfempl"
```

- **Number of terminal nodes: 13**
- **Residual mean deviance: 0.05204 = 46.16 / 887**
- **Misclassification error rate: 0.01444 = 13 / 900**

Tree Table

```
node), split, n, deviance, yval, (yprob)
      * denotes terminal node
  1) root 900 970.700 1 ( 0.2300 0.770000 )
    2) Avgexp<0.46 221 104.300 0 ( 0.9367 0.063350 )
8) ActiveCard<1.5 63  17.740 0 ( 0.9683 0.031750 )
         16) Income<2.8134 41   0.000 0 ( 1.0000 0.000000 ) *
         17) Income>2.8134 22  13.400 0 ( 0.9091 0.090910 )
           34) Income<3.3314 5   6.730 0 ( 0.6000 0.400000 ) *
           35) Income>3.3314 17   0.000 0 ( 1.0000 0.000000 ) *
        9) ActiveCard>1.5 47  51.150 0 ( 0.7660 0.234000 )
         18) Dependent<0.5 27  34.370 0 ( 0.6667 0.333300 )
           36) Income<2.67 18  19.070 0 ( 0.7778 0.222200 )
             72) Income<2.467 13  16.050 0 ( 0.6923 0.307700 )
              144) ActiveCard<5 7   5.742 0 ( 0.8571 0.142900 ) *
              145) ActiveCard>5 6   8.318 0 ( 0.5000 0.500000 ) *
             73) Income>2.467 5   0.000 0 ( 1.0000 0.000000 ) *
           37) Income>2.67 9  12.370 1 ( 0.4444 0.555600 ) *
         19) Dependent>0.5 20  13.000 0 ( 0.9000 0.100000 )
           38) Income<2.9 10  10.010 0 ( 0.8000 0.200000 )
             76) Income<1.9545 5   0.000 0 ( 1.0000 0.000000 ) *
             77) Income>1.9545 5   6.730 0 ( 0.6000 0.400000 ) *
           39) Income>2.9 10   0.000 0 ( 1.0000 0.000000 ) *
      5) Major>0.5 111  11.410 0 ( 0.9910 0.009009 )
       10) Selfempl:0 102   0.000 0 ( 1.0000 0.000000 ) *
       11) Selfempl:1 9   6.279 0 ( 0.8889 0.111100 ) *
    3) Avgexp>0.46 679   0.000 1 ( 0.0000 1.000000 ) *
```

The graphic tree



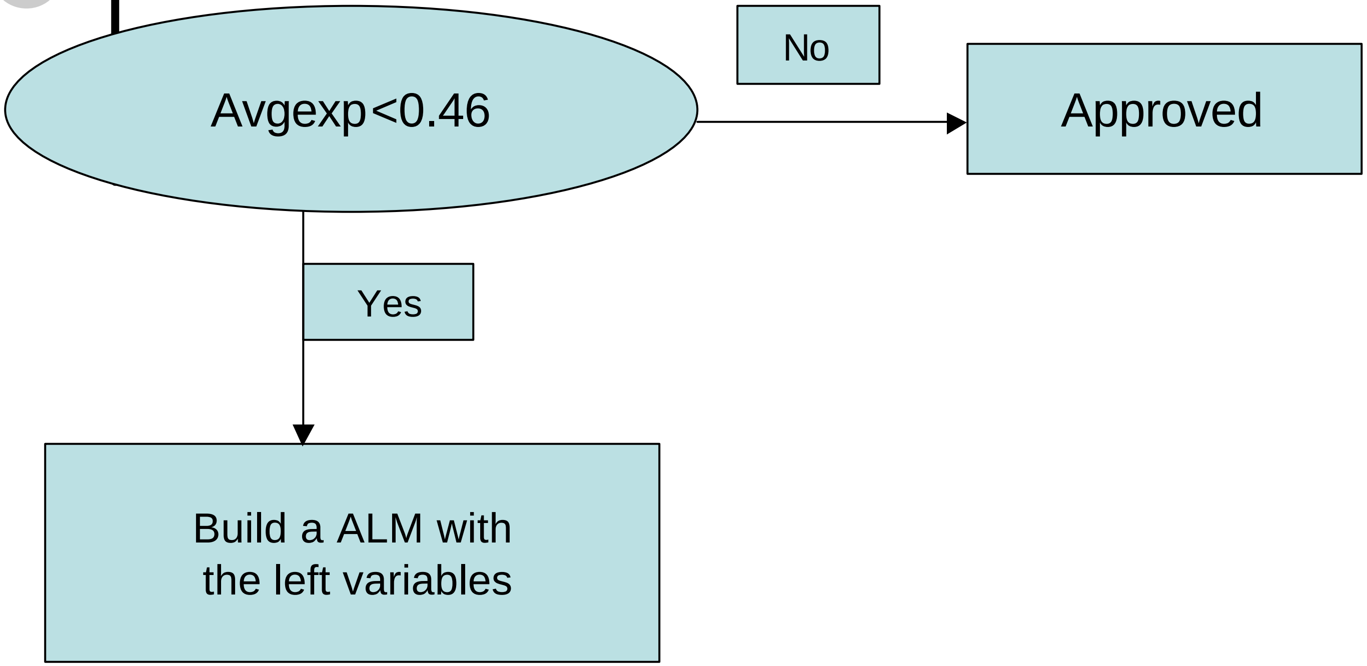
Prediction based on test data:

```
predict(object = card.tree, newdata = test.card, type = "tree")

Variables actually used in tree construction:

[1] "Avgexp"     "Major"      "ActiveCard" "Income" "Dependent"

[6] "Selfempl"

Number of terminal nodes:  13

Residual mean deviance:  0.1903 = 77.27 / 406

Misclassification error rate: 0.01909 = 8 / 419
```

Results

- Avgexp --an important factor in explaining the response.
- If a person spends more than 4600 dollars a month with a credit card, he will get his application for a new credit card approved.
- Explore other important factors that will affect credit card company’s decision for those people who don’t have a credit card or who never uses a credit card even if he has one and thus with a monthly expenditure less than 460 dollars. Try our second method logistic additive model for the subset of data!

What does Classification Tree tell us?



Avgexp<0.46

No

Approved

Yes

Build a ALM with the left variables

First Step: Interactions?

Fit a logistic linear model with these two interactions

	Value	Std. Error	t value
(Intercept)	-4.1302777733	1.454052564	-2.8405285
Major	-4.1734938593	4.261504253	-0.9793476
Activecard	0.0872526190	0.042670080	**2.0448197**
Age	0.0312218743	0.027603088	1.1311008
Selfempl	0.6196866490	0.927336901	0.6682433
Income	-0.0377325619	0.282275008	-0.1336731
Dependent	-0.8781425957	0.443802604	-1.9786783
Ownrent	-0.2134269692	0.862998289	-0.2473087
Curadd	0.0037512577	0.008211349	0.4568382
Majorcard	1.3154680105	1.093865429	1.2025867
Major:Age	0.0452595301	0.085880177	**0.5270079**
Curadd:Income	-0.0006090426	0.001399399	**-0.4352173**

build a logistic linear model without interactions

```
                   Value  Std. Error    t value
(Intercept) -3.996717756 1.377657126 -2.9010976
      Major -2.172780323 1.012372466 -2.1462262
 Activecard  0.090005814 0.042780360  2.103905
        Age  0.035067755 0.026921032  1.3026156
   Selfempl  0.574306148 0.944125907  0.6082940
     Income -0.135455821 0.207910355 -0.6515107
  Dependent -0.867295066 0.442612728 -1.9594897
    Ownrent -0.171289519 0.846023074 -0.2024644
     Curadd  0.001240073 0.004633245  0.2676467
  Majorcard  1.322985273 1.094882289  1.2083356
```

```
(Dispersion Parameter for Binomial family taken to be 1 )
Null Deviance: 104.3487 on 220 degrees of freedom
Residual Deviance: 73.87756 on 211 degrees of freedom
Number of Fisher Scoring Iterations: 9
```

Second Step: we build a Logistic Additive model to check the nonlinear part of each predictor.

```
                  Df Npar Df Npar Chisq   P(Chi)
   (Intercept)  1
      s(Major)  1     0.9    0.02540 0.8247796
 s(Activecard)  1     2.9    5.33693 0.1402896
        s(Age)  1     3.0   11.12546 0.0107119
      Selfempl  1
     s(Income)  1     2.9    3.88296 0.2610870
  s(Dependent)  1     1.9    0.23180 0.8677210
       Ownrent  1
      s(Curadd) 1     3.0    1.46328 0.6842343
     Majorcard  1

 Null Deviance: 104.3487 on 220 degrees of freedom
Residual Deviance: 47.44631 on 196.5519 degrees of freedom
Number of Local Scoring Iterations: 16
DF for Terms and Chi-squares for Nonparametric Effects
```

Third Step: we choose the important factors to fit another model with linear part of Major and Activecard, and nonlinear of Age.

```
              Df Npar Df Npar Chisq    P(Chi)
  (Intercept)  1
     s(Major)  1     0.8    0.02161 0.8316526
s(Activecard)  1     2.9    4.15372 0.2382494
       s(Age)  1     2.8   12.14654 0.0058192
```

```
Null Deviance: 104.3487 on 220 degrees of freedom
Residual Deviance: 64.12138 on 210.3879 degrees of freedom
Number of Local Scoring Iterations: 16
DF for Terms and Chi-squares for Nonparametric Effects
```

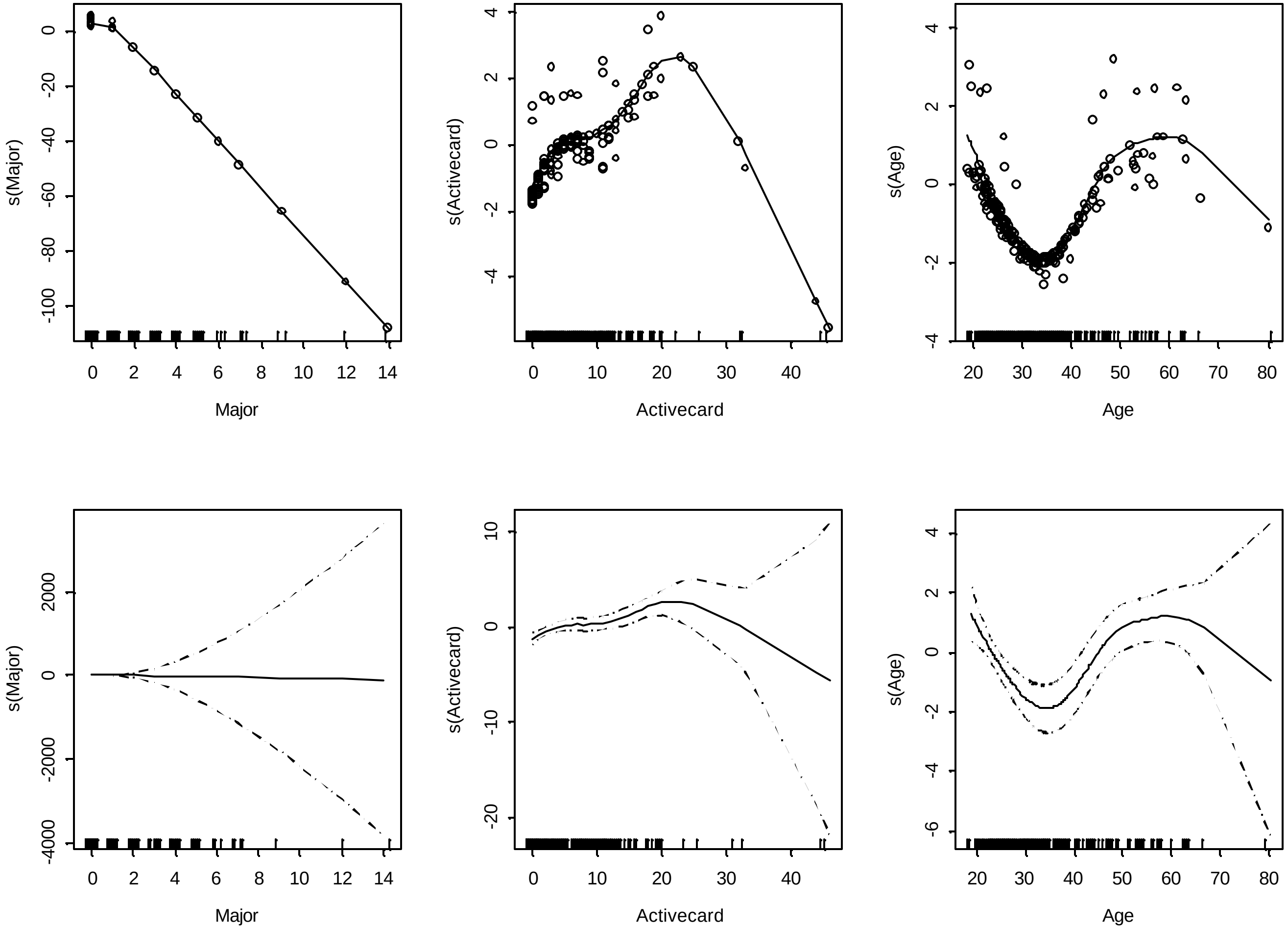


Figure: The partial fits for the ALM

Fourth Step: we compare the two models, one is the logistic additive model with all the variables and the other with only Major, Activecard and Age.

```
Response: Approval
          Terms
1 s(Major) + s(Activecard) + s(Age)
2 s(Major) + s(Activecard) + s(Age) + Selfempl +
s(Income) + s(Dependent) + Ownrent + s(Curadd) + Majorcard
  Resid. Df Resid. Dev
1  210.3879   64.12137
2  196.5519   47.44631
                          Test       Df
1
2 +Selfempl+s(Income)+s(Dependent)+Ownrent+s(Curadd)+Majorcard 13.83607
  Deviance   Pr(Chi)
1
2 16.67507 0.2637644
```

Fifth Step: Since the effects of Major and Activecard on the response are linear, and Age is nonlinear, and some quadratic, we build a model with a second degree polynomial for Age, and linear for Major and Activecard.

```
                        Value Std. Error    t value
 (Intercept)       -2.43028340 0.38150902 -6.3701860
      Major        -2.15570628 1.03787786 -2.0770327
  Activecard        0.07113605 0.03586683  1.9833378
poly(Age, 2)1       3.77684780 3.67773677  1.0269489
poly(Age, 2)2       3.07078526 3.16378516  0.9706049

(Dispersion Parameter for Binomial family taken to be 1 )
Null Deviance: 104.3487 on 220 degrees of freedom
Residual Deviance: 82.47548 on 216 degrees of freedom
Number of Fisher Scoring Iterations: 9
```

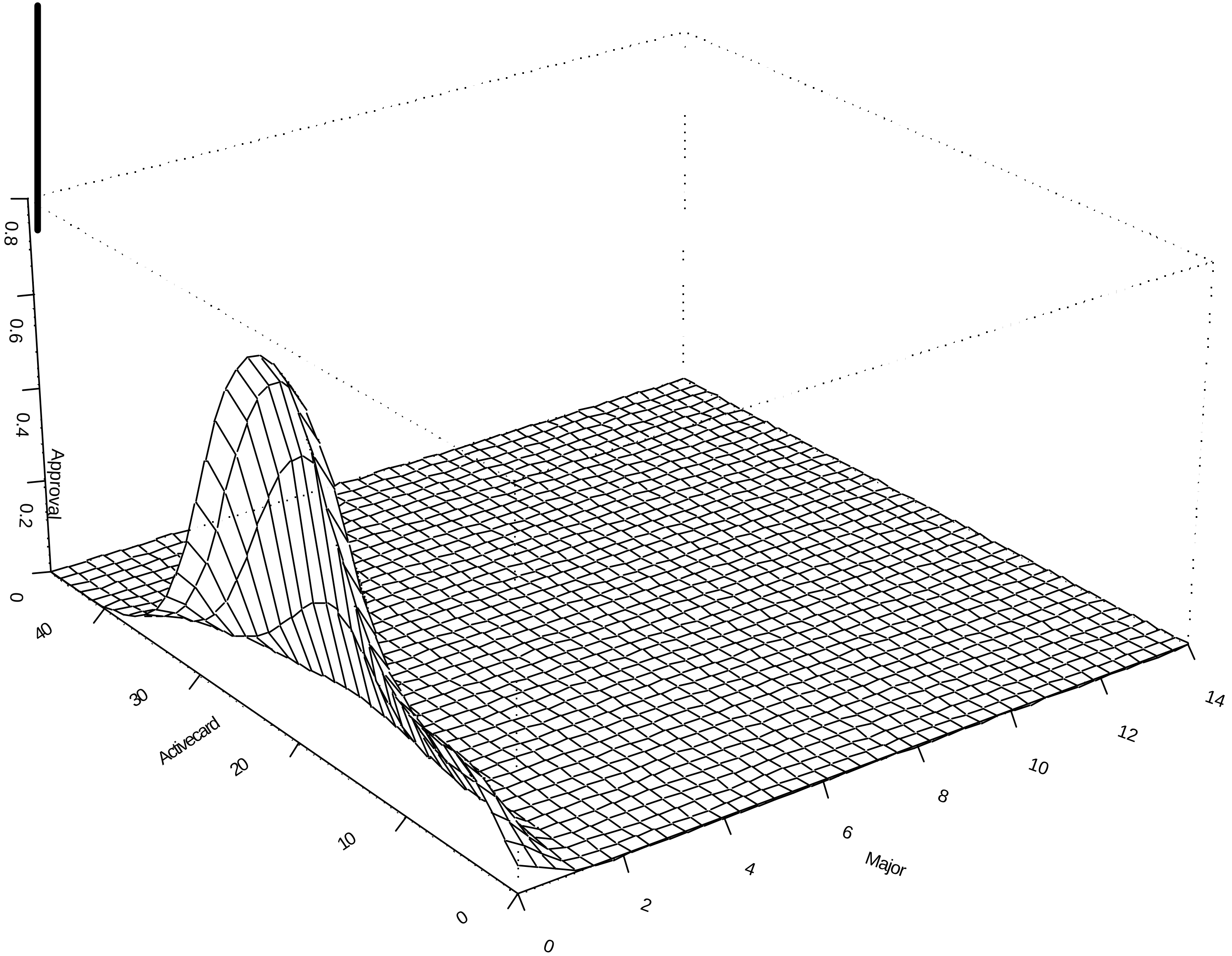
we can compare these two models. The results indicate that the logistic additive model is much better than the linear model.

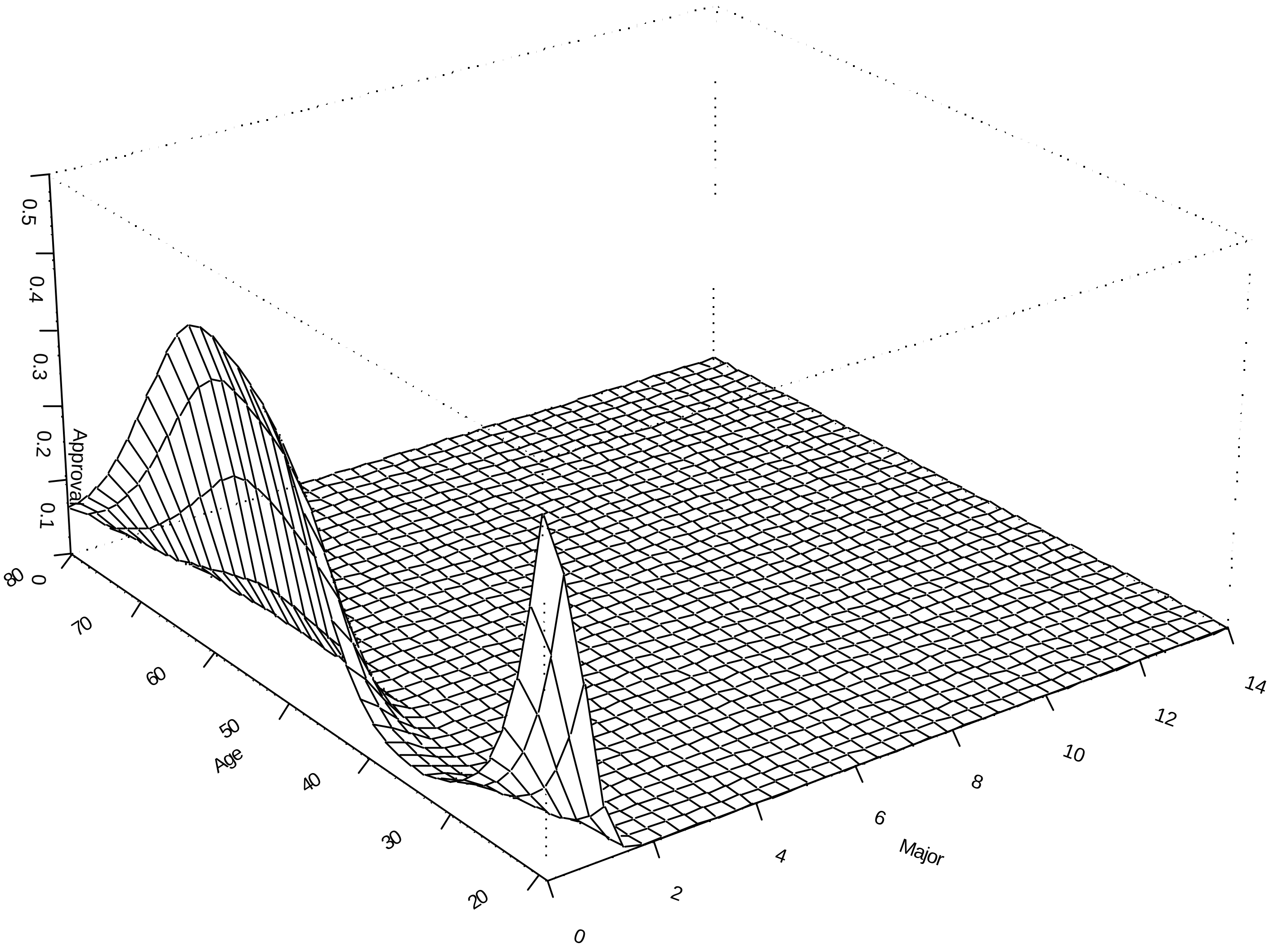
```
                                     Terms Resid. Df Resid. Dev    Test       Df
1 Major + Activecard + poly(Age, 2)  216.0000   82.47548
2 s(Major) + s(Activecard) + s(Age)  210.3879   64.12137 1 vs. 2
    5.612069
  Deviance    Pr(Chi)
1
2 18.35411 0.00407996
```

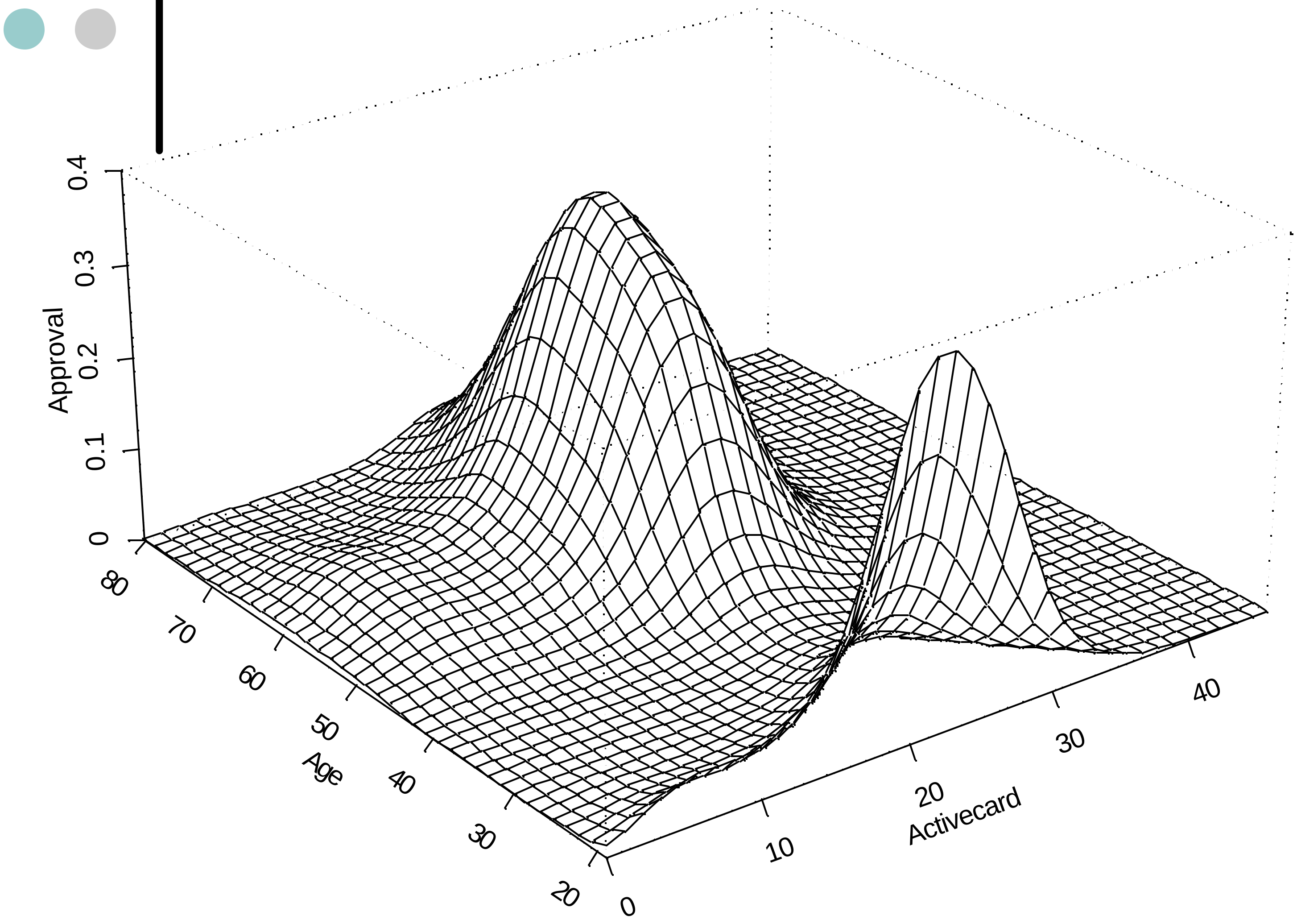
Predicting the Logistic Additive Model

The overall training error is 0.0167

The overall testing error is .00968.


Approval
0.8
0.6
0.4
0.2
0
Activecard
40
30
20
10
0
Major
0
2
4
6
8
10
12
14


Approval
0.5
0.4
0.3
0.2
0.1
0
Age
80
70
60
50
40
30
20
Major
0
2
4
6
8
10
12
14



Approval
0.4
0.3
0.2
0.1
0
80
70
60
50
Age
40
30
20
0
10
20
Activecard
30
40

Statistical Classifiers
Support Vector Machine

- Support vector machines (SVMs) are a new generation of learning system. It is based on strong mathematical fundations (the statistical learning theory developed by Vladimir Vapnik since the 70's) and results in simple yet very powerful algorithms
- Support vector machine uses theories from optimization and statistical theory and combines these two in support vector machine.

C-classification

- For this type of SVM, training involves the minimization of the error function:

$$\frac{1}{2} w^T w + C\sum_{i=1}^{N} x_i$$

- subject to the constraints:

$$y_i(w^T f(x_i) + b \geq 1 - x_i \ and\, x_i \geq 0, i = 1,\ldots, N$$

- where C is the capacity constant, w is the vector of coefficients, b a constant and these are parameters for handling nonseparable data (inputs). The index i labels the N training cases. Note that it is the class label and xi’s are the independent variables. The kernel is used to transform data from the input (independent) to the feature space. It should be noted that the larger the C, the more the error is penalized. Thus, C should be chosen with care to avoid over fitting.

Model summary

The parameters are used in the model

Gamma =0.1, C=1, error =.1044

Gamma=0.125, C=16, error=.0923

Since the smaller the C, the less the error is penalized, we would like to use

Gamma =0.1, C=1, error =.1044

Test result

Gamma=0.1
C=1
	Training Data	Testing Data

Error	0.1044	0.1241

Conclusion

	Classification Tree	ALM	SVM
Training Error Rate	0.01444	0.0167	0.1044
Testing Error Rate	0.01909	0.00968	0.1241

Table: Comparison of training error rate and testing error rate of Three Methods