

# Intro to Data Science - Lab 9

Copyright 2022, Jeffrey Stanton and Jeffrey Saltz Please do not post online.

## Week 9 - Supervised Data Mining

*# Enter your name here: Taylor Huang*

Please include nice comments.

### Instructions:

Run the necessary code on your own instance of R-Studio.

**Attribution statement: (choose only one and delete the rest)**

**Supervised data mining/machine learning** is the most prevalent form of data mining as it allows for the prediction of new cases in the future. For example, when credit card companies are trying to detect fraud, they will create a supervised model by training it on fraud data that they already have. Then they will deploy the model into the field: As new input data arrives the model predicts whether it seems fraudulent and flags those transactions where that probability is high.

In these exercises we will work with a built-in data set called **GermanCredit**. This data set is in the **caret** package so we will need that and the **kernlab** package to be installed and libraried before running the following:

# 1. I did this lab assignment by myself, with help from the book and the professor.

```
```r
#install.packages(caret)
library("caret")

## Loading required package: ggplot2

## Loading required package: lattice

#install.packages(kernlab)
library("kernlab")

##
## Attaching package: 'kernlab'

## The following object is masked from 'package:ggplot2':
##
##      alpha

data("GermanCredit")
subCredit <- GermanCredit[,1:10]
str(subCredit)

## 'data.frame':    1000 obs. of  10 variables:
##  $ Duration      : int  6 48 12 42 24 36 24 36 12 30 ...
##  $ Amount        : int  1169 5951 2096 7882 4870 9055 2835 6948 3059 5234 ...
##  $ InstallmentRatePercentage: int  4 2 2 2 3 2 3 2 2 4 ...
##  $ ResidenceDuration : int  4 2 3 4 4 4 4 2 4 2 ...
##  $ Age           : int  67 22 49 45 53 35 53 35 61 28 ...
##  $ NumberExistingCredits : int  2 1 1 1 2 1 1 1 1 2 ...
##  $ NumberPeopleMaintenance : int  1 1 2 2 2 2 1 1 1 1 ...
##  $ Telephone      : num  0 1 1 1 1 0 1 0 1 1 ...
```

```
## $ ForeignWorker      : num  1 1 1 1 1 1 1 1 1 1 ...
## $ Class              : Factor w/ 2 levels "Bad","Good": 2 1 2 2 1 2 2 2 1 ...
```

1. Examine the data structure that `str()` reveals. Also use the `help()` command to learn more about the **GermanCredit** data set. Summarize what you see in a comment.

```
help(GermanCredit)
```

*#In subCredit, you see different entries related to individuals' credit data in Germany. subCredit also*

2. Use the `createDataPartition()` function to generate a list of cases to include in the training data. This function is conveniently provided by `caret` and allows one to directly control the number of training cases. It also ensures that the training cases are balanced with respect to the outcome variable. Try this: `trainList <- createDataPartition(y=subCredit$Class,p=.40,list=FALSE)`

```
trainList <- createDataPartition(y=subCredit$Class,p=.40,list=FALSE)
trainList
```

```
##      Resample1
## [1,]         1
## [2,]         8
## [3,]        18
## [4,]        19
## [5,]        20
## [6,]        25
## [7,]        26
## [8,]        28
## [9,]        29
## [10,]       30
## [11,]       35
## [12,]       39
## [13,]       41
## [14,]       44
## [15,]       46
## [16,]       47
## [17,]       48
## [18,]       49
## [19,]       53
## [20,]       54
## [21,]       56
## [22,]       57
## [23,]       61
## [24,]       62
## [25,]       63
## [26,]       64
## [27,]       65
## [28,]       67
## [29,]       73
## [30,]       74
## [31,]       75
## [32,]       76
## [33,]       79
## [34,]       80
## [35,]       82
## [36,]       87
## [37,]       89
## [38,]       94
```

##	[39,]	95
##	[40,]	105
##	[41,]	107
##	[42,]	108
##	[43,]	112
##	[44,]	113
##	[45,]	114
##	[46,]	116
##	[47,]	122
##	[48,]	124
##	[49,]	132
##	[50,]	133
##	[51,]	138
##	[52,]	140
##	[53,]	143
##	[54,]	144
##	[55,]	148
##	[56,]	149
##	[57,]	152
##	[58,]	161
##	[59,]	162
##	[60,]	163
##	[61,]	164
##	[62,]	169
##	[63,]	172
##	[64,]	176
##	[65,]	179
##	[66,]	182
##	[67,]	183
##	[68,]	185
##	[69,]	186
##	[70,]	188
##	[71,]	189
##	[72,]	192
##	[73,]	193
##	[74,]	196
##	[75,]	204
##	[76,]	205
##	[77,]	208
##	[78,]	209
##	[79,]	210
##	[80,]	212
##	[81,]	213
##	[82,]	217
##	[83,]	218
##	[84,]	222
##	[85,]	224
##	[86,]	227
##	[87,]	228
##	[88,]	236
##	[89,]	237
##	[90,]	239
##	[91,]	240
##	[92,]	241

##	[93,]	243
##	[94,]	246
##	[95,]	248
##	[96,]	249
##	[97,]	250
##	[98,]	251
##	[99,]	252
##	[100,]	257
##	[101,]	258
##	[102,]	266
##	[103,]	268
##	[104,]	270
##	[105,]	271
##	[106,]	275
##	[107,]	277
##	[108,]	281
##	[109,]	282
##	[110,]	283
##	[111,]	284
##	[112,]	285
##	[113,]	287
##	[114,]	288
##	[115,]	293
##	[116,]	298
##	[117,]	303
##	[118,]	304
##	[119,]	305
##	[120,]	309
##	[121,]	310
##	[122,]	311
##	[123,]	312
##	[124,]	313
##	[125,]	314
##	[126,]	316
##	[127,]	320
##	[128,]	321
##	[129,]	324
##	[130,]	325
##	[131,]	326
##	[132,]	329
##	[133,]	333
##	[134,]	336
##	[135,]	340
##	[136,]	341
##	[137,]	342
##	[138,]	343
##	[139,]	346
##	[140,]	350
##	[141,]	352
##	[142,]	353
##	[143,]	356
##	[144,]	357
##	[145,]	358
##	[146,]	359

## [147,]	363
## [148,]	364
## [149,]	367
## [150,]	369
## [151,]	376
## [152,]	380
## [153,]	381
## [154,]	382
## [155,]	383
## [156,]	384
## [157,]	386
## [158,]	387
## [159,]	390
## [160,]	391
## [161,]	396
## [162,]	402
## [163,]	404
## [164,]	412
## [165,]	420
## [166,]	421
## [167,]	426
## [168,]	427
## [169,]	428
## [170,]	429
## [171,]	432
## [172,]	433
## [173,]	434
## [174,]	437
## [175,]	438
## [176,]	440
## [177,]	442
## [178,]	443
## [179,]	444
## [180,]	446
## [181,]	447
## [182,]	448
## [183,]	454
## [184,]	457
## [185,]	459
## [186,]	462
## [187,]	463
## [188,]	465
## [189,]	467
## [190,]	468
## [191,]	469
## [192,]	470
## [193,]	474
## [194,]	479
## [195,]	481
## [196,]	483
## [197,]	486
## [198,]	488
## [199,]	491
## [200,]	492

## [201,]	493
## [202,]	497
## [203,]	499
## [204,]	501
## [205,]	502
## [206,]	503
## [207,]	506
## [208,]	508
## [209,]	514
## [210,]	525
## [211,]	526
## [212,]	527
## [213,]	533
## [214,]	535
## [215,]	537
## [216,]	539
## [217,]	540
## [218,]	543
## [219,]	545
## [220,]	549
## [221,]	555
## [222,]	556
## [223,]	557
## [224,]	560
## [225,]	561
## [226,]	562
## [227,]	563
## [228,]	564
## [229,]	568
## [230,]	569
## [231,]	573
## [232,]	576
## [233,]	578
## [234,]	579
## [235,]	581
## [236,]	582
## [237,]	583
## [238,]	586
## [239,]	591
## [240,]	592
## [241,]	593
## [242,]	594
## [243,]	595
## [244,]	597
## [245,]	599
## [246,]	600
## [247,]	603
## [248,]	604
## [249,]	608
## [250,]	609
## [251,]	612
## [252,]	613
## [253,]	614
## [254,]	615

## [255,]	616
## [256,]	622
## [257,]	623
## [258,]	633
## [259,]	635
## [260,]	637
## [261,]	641
## [262,]	643
## [263,]	645
## [264,]	648
## [265,]	651
## [266,]	655
## [267,]	657
## [268,]	659
## [269,]	660
## [270,]	664
## [271,]	665
## [272,]	672
## [273,]	677
## [274,]	679
## [275,]	680
## [276,]	681
## [277,]	683
## [278,]	684
## [279,]	685
## [280,]	686
## [281,]	689
## [282,]	690
## [283,]	692
## [284,]	698
## [285,]	700
## [286,]	701
## [287,]	705
## [288,]	708
## [289,]	714
## [290,]	715
## [291,]	716
## [292,]	717
## [293,]	721
## [294,]	723
## [295,]	726
## [296,]	728
## [297,]	729
## [298,]	730
## [299,]	732
## [300,]	735
## [301,]	737
## [302,]	740
## [303,]	742
## [304,]	743
## [305,]	745
## [306,]	746
## [307,]	747
## [308,]	750

## [309,]	763
## [310,]	764
## [311,]	765
## [312,]	770
## [313,]	772
## [314,]	774
## [315,]	776
## [316,]	778
## [317,]	787
## [318,]	794
## [319,]	795
## [320,]	796
## [321,]	797
## [322,]	798
## [323,]	799
## [324,]	800
## [325,]	802
## [326,]	803
## [327,]	804
## [328,]	806
## [329,]	812
## [330,]	815
## [331,]	816
## [332,]	820
## [333,]	821
## [334,]	822
## [335,]	823
## [336,]	824
## [337,]	828
## [338,]	829
## [339,]	830
## [340,]	832
## [341,]	834
## [342,]	835
## [343,]	837
## [344,]	839
## [345,]	840
## [346,]	841
## [347,]	842
## [348,]	844
## [349,]	846
## [350,]	852
## [351,]	853
## [352,]	855
## [353,]	865
## [354,]	866
## [355,]	867
## [356,]	871
## [357,]	872
## [358,]	873
## [359,]	874
## [360,]	876
## [361,]	878
## [362,]	881



```
## [363,]      882
## [364,]      894
## [365,]      903
## [366,]      908
## [367,]      913
## [368,]      915
## [369,]      916
## [370,]      917
## [371,]      920
## [372,]      921
## [373,]      925
## [374,]      928
## [375,]      929
## [376,]      932
## [377,]      933
## [378,]      938
## [379,]      941
## [380,]      945
## [381,]      946
## [382,]      947
## [383,]      949
## [384,]      951
## [385,]      952
## [386,]      955
## [387,]      956
## [388,]      957
## [389,]      958
## [390,]      959
## [391,]      963
## [392,]      965
## [393,]      969
## [394,]      971
## [395,]      973
## [396,]      983
## [397,]      987
## [398,]      993
## [399,]      996
## [400,]     1000
```

3. Examine the contents of **trainList** to make sure that it is a list of case numbers. With **p=0.40**, it should have 400 case numbers in it.

```
# it is in a list of numbers with 400 cases
```

4. What is **trainList**? What do the elements in **trainList** represent? Which attribute is balanced in the **trainList** dataset?

```
# The elements of TrainList represents the people who have poor class, and the count of these people.
```

5. Use **trainList** and the square brackets notation to create a training data set called **\*\* trainSet \*\*** from the **subCredit** data frame. Look at the structure of **trainSet** to make sure it has all of the same variables as **subCredit**. The **trainSet** structure should be a data frame with **400 rows and 10 columns**.

```
trainSet <- subCredit[trainList,]
trainList
```

##	Resample1
## [1,]	1
## [2,]	8
## [3,]	18
## [4,]	19
## [5,]	20
## [6,]	25
## [7,]	26
## [8,]	28
## [9,]	29
## [10,]	30
## [11,]	35
## [12,]	39
## [13,]	41
## [14,]	44
## [15,]	46
## [16,]	47
## [17,]	48
## [18,]	49
## [19,]	53
## [20,]	54
## [21,]	56
## [22,]	57
## [23,]	61
## [24,]	62
## [25,]	63
## [26,]	64
## [27,]	65
## [28,]	67
## [29,]	73
## [30,]	74
## [31,]	75
## [32,]	76
## [33,]	79
## [34,]	80
## [35,]	82
## [36,]	87
## [37,]	89
## [38,]	94
## [39,]	95
## [40,]	105
## [41,]	107
## [42,]	108
## [43,]	112
## [44,]	113
## [45,]	114
## [46,]	116
## [47,]	122
## [48,]	124
## [49,]	132
## [50,]	133
## [51,]	138
## [52,]	140
## [53,]	143

##	[54,]	144
##	[55,]	148
##	[56,]	149
##	[57,]	152
##	[58,]	161
##	[59,]	162
##	[60,]	163
##	[61,]	164
##	[62,]	169
##	[63,]	172
##	[64,]	176
##	[65,]	179
##	[66,]	182
##	[67,]	183
##	[68,]	185
##	[69,]	186
##	[70,]	188
##	[71,]	189
##	[72,]	192
##	[73,]	193
##	[74,]	196
##	[75,]	204
##	[76,]	205
##	[77,]	208
##	[78,]	209
##	[79,]	210
##	[80,]	212
##	[81,]	213
##	[82,]	217
##	[83,]	218
##	[84,]	222
##	[85,]	224
##	[86,]	227
##	[87,]	228
##	[88,]	236
##	[89,]	237
##	[90,]	239
##	[91,]	240
##	[92,]	241
##	[93,]	243
##	[94,]	246
##	[95,]	248
##	[96,]	249
##	[97,]	250
##	[98,]	251
##	[99,]	252
##	[100,]	257
##	[101,]	258
##	[102,]	266
##	[103,]	268
##	[104,]	270
##	[105,]	271
##	[106,]	275
##	[107,]	277

## [108,]	281
## [109,]	282
## [110,]	283
## [111,]	284
## [112,]	285
## [113,]	287
## [114,]	288
## [115,]	293
## [116,]	298
## [117,]	303
## [118,]	304
## [119,]	305
## [120,]	309
## [121,]	310
## [122,]	311
## [123,]	312
## [124,]	313
## [125,]	314
## [126,]	316
## [127,]	320
## [128,]	321
## [129,]	324
## [130,]	325
## [131,]	326
## [132,]	329
## [133,]	333
## [134,]	336
## [135,]	340
## [136,]	341
## [137,]	342
## [138,]	343
## [139,]	346
## [140,]	350
## [141,]	352
## [142,]	353
## [143,]	356
## [144,]	357
## [145,]	358
## [146,]	359
## [147,]	363
## [148,]	364
## [149,]	367
## [150,]	369
## [151,]	376
## [152,]	380
## [153,]	381
## [154,]	382
## [155,]	383
## [156,]	384
## [157,]	386
## [158,]	387
## [159,]	390
## [160,]	391
## [161,]	396

## [162,]	402
## [163,]	404
## [164,]	412
## [165,]	420
## [166,]	421
## [167,]	426
## [168,]	427
## [169,]	428
## [170,]	429
## [171,]	432
## [172,]	433
## [173,]	434
## [174,]	437
## [175,]	438
## [176,]	440
## [177,]	442
## [178,]	443
## [179,]	444
## [180,]	446
## [181,]	447
## [182,]	448
## [183,]	454
## [184,]	457
## [185,]	459
## [186,]	462
## [187,]	463
## [188,]	465
## [189,]	467
## [190,]	468
## [191,]	469
## [192,]	470
## [193,]	474
## [194,]	479
## [195,]	481
## [196,]	483
## [197,]	486
## [198,]	488
## [199,]	491
## [200,]	492
## [201,]	493
## [202,]	497
## [203,]	499
## [204,]	501
## [205,]	502
## [206,]	503
## [207,]	506
## [208,]	508
## [209,]	514
## [210,]	525
## [211,]	526
## [212,]	527
## [213,]	533
## [214,]	535
## [215,]	537

## [216,]	539
## [217,]	540
## [218,]	543
## [219,]	545
## [220,]	549
## [221,]	555
## [222,]	556
## [223,]	557
## [224,]	560
## [225,]	561
## [226,]	562
## [227,]	563
## [228,]	564
## [229,]	568
## [230,]	569
## [231,]	573
## [232,]	576
## [233,]	578
## [234,]	579
## [235,]	581
## [236,]	582
## [237,]	583
## [238,]	586
## [239,]	591
## [240,]	592
## [241,]	593
## [242,]	594
## [243,]	595
## [244,]	597
## [245,]	599
## [246,]	600
## [247,]	603
## [248,]	604
## [249,]	608
## [250,]	609
## [251,]	612
## [252,]	613
## [253,]	614
## [254,]	615
## [255,]	616
## [256,]	622
## [257,]	623
## [258,]	633
## [259,]	635
## [260,]	637
## [261,]	641
## [262,]	643
## [263,]	645
## [264,]	648
## [265,]	651
## [266,]	655
## [267,]	657
## [268,]	659
## [269,]	660

## [270,]	664
## [271,]	665
## [272,]	672
## [273,]	677
## [274,]	679
## [275,]	680
## [276,]	681
## [277,]	683
## [278,]	684
## [279,]	685
## [280,]	686
## [281,]	689
## [282,]	690
## [283,]	692
## [284,]	698
## [285,]	700
## [286,]	701
## [287,]	705
## [288,]	708
## [289,]	714
## [290,]	715
## [291,]	716
## [292,]	717
## [293,]	721
## [294,]	723
## [295,]	726
## [296,]	728
## [297,]	729
## [298,]	730
## [299,]	732
## [300,]	735
## [301,]	737
## [302,]	740
## [303,]	742
## [304,]	743
## [305,]	745
## [306,]	746
## [307,]	747
## [308,]	750
## [309,]	763
## [310,]	764
## [311,]	765
## [312,]	770
## [313,]	772
## [314,]	774
## [315,]	776
## [316,]	778
## [317,]	787
## [318,]	794
## [319,]	795
## [320,]	796
## [321,]	797
## [322,]	798
## [323,]	799

## [324,]	800
## [325,]	802
## [326,]	803
## [327,]	804
## [328,]	806
## [329,]	812
## [330,]	815
## [331,]	816
## [332,]	820
## [333,]	821
## [334,]	822
## [335,]	823
## [336,]	824
## [337,]	828
## [338,]	829
## [339,]	830
## [340,]	832
## [341,]	834
## [342,]	835
## [343,]	837
## [344,]	839
## [345,]	840
## [346,]	841
## [347,]	842
## [348,]	844
## [349,]	846
## [350,]	852
## [351,]	853
## [352,]	855
## [353,]	865
## [354,]	866
## [355,]	867
## [356,]	871
## [357,]	872
## [358,]	873
## [359,]	874
## [360,]	876
## [361,]	878
## [362,]	881
## [363,]	882
## [364,]	894
## [365,]	903
## [366,]	908
## [367,]	913
## [368,]	915
## [369,]	916
## [370,]	917
## [371,]	920
## [372,]	921
## [373,]	925
## [374,]	928
## [375,]	929
## [376,]	932
## [377,]	933



```
## [378,]      938
## [379,]      941
## [380,]      945
## [381,]      946
## [382,]      947
## [383,]      949
## [384,]      951
## [385,]      952
## [386,]      955
## [387,]      956
## [388,]      957
## [389,]      958
## [390,]      959
## [391,]      963
## [392,]      965
## [393,]      969
## [394,]      971
## [395,]      973
## [396,]      983
## [397,]      987
## [398,]      993
## [399,]      996
## [400,]     1000
```

6. Use **trainList** and the square brackets notation to create a testing data set called **testSet** from the **subCredit** data frame. The **testSet** structure should be a data frame with **600 rows and 10 columns** and should be a completely different set of cases than **trainSet**.

```
testSet <- subCredit[-trainList,]
testSet
```

```
##      Duration Amount InstallmentRatePercentage ResidenceDuration Age
## 2           48   5951                    2              2  22
## 3           12   2096                    2              3  49
## 4           42   7882                    2              4  45
## 5           24   4870                    3              4  53
## 6           36   9055                    2              4  35
## 7           24   2835                    3              4  53
## 9           12   3059                    2              4  61
## 10          30   5234                    4              2  28
## 11          12   1295                    3              1  25
## 12          48   4308                    3              4  24
## 13          12   1567                    1              1  22
## 14          24   1199                    4              4  60
## 15          15   1403                    2              4  28
## 16          24   1282                    4              2  32
## 17          24   2424                    4              4  53
## 21           9   2134                    4              4  48
## 22           6   2647                    2              3  44
## 23          10   2241                    1              3  48
## 24          12   1804                    3              4  44
## 27           6    426                    4              4  39
## 31          18   1913                    3              3  36
## 32          24   4020                    2              2  27
## 33          18   5866                    2              2  30
```

## 34	12	1264	4	4	57
## 36	45	4746	4	2	25
## 37	48	6110	1	3	31
## 38	18	2100	4	2	37
## 40	9	458	4	3	24
## 42	12	1158	3	1	26
## 43	18	6204	2	4	44
## 45	48	6143	4	4	58
## 50	12	2073	4	2	28
## 51	24	2333	4	2	29
## 52	27	5965	1	2	30
## 55	36	2225	4	4	57
## 58	36	9566	2	2	31
## 59	18	1961	3	2	23
## 60	36	6229	4	4	23
## 66	27	5190	4	4	48
## 68	12	1007	4	1	22
## 69	36	1819	4	4	37
## 70	36	2394	4	4	25
## 71	36	8133	1	2	30
## 72	7	730	4	2	46
## 77	42	3965	4	3	34
## 78	11	4771	2	4	51
## 81	24	5943	1	1	44
## 83	18	1568	3	4	24
## 84	24	1755	4	4	58
## 85	10	2315	3	4	52
## 86	12	1412	4	2	29
## 88	36	12612	1	4	47
## 90	12	1108	4	3	28
## 91	12	618	4	4	56
## 92	12	1409	4	3	54
## 93	12	797	4	3	33
## 96	54	15945	3	4	58
## 97	12	2012	4	2	61
## 98	18	2622	4	4	34
## 99	36	2337	4	4	36
## 100	20	7057	3	4	36
## 101	24	1469	4	4	41
## 102	36	2323	4	4	24
## 103	6	932	3	2	24
## 104	9	1919	4	3	35
## 106	24	11938	2	3	39
## 109	24	7721	1	2	30
## 110	14	1410	1	2	35
## 111	6	1449	1	2	31
## 115	12	1680	3	1	35
## 117	42	7174	4	3	30
## 118	10	2132	2	3	27
## 119	33	4281	1	4	23
## 120	12	2366	3	3	36
## 121	21	1835	3	2	25
## 123	12	1768	3	2	24
## 125	18	1924	4	3	27

## 126	12	2121	4	2	30
## 127	12	701	4	2	40
## 128	12	639	4	2	30
## 129	12	1860	4	2	34
## 130	12	3499	3	2	29
## 131	48	8487	1	2	24
## 134	18	1984	4	4	47
## 135	60	10144	2	4	21
## 136	12	1240	4	2	38
## 137	27	8613	2	2	27
## 139	15	2728	4	2	35
## 141	6	709	2	2	27
## 142	36	4795	4	1	30
## 145	21	2288	4	4	23
## 146	48	3566	4	2	30
## 147	6	860	1	4	39
## 150	18	1582	4	4	46
## 151	6	1346	2	4	42
## 153	36	5848	4	1	24
## 154	24	7758	2	4	29
## 155	24	6967	4	4	36
## 156	12	1282	2	4	20
## 157	9	1288	3	4	48
## 158	12	339	4	1	45
## 159	24	3512	2	3	38
## 160	6	1898	1	2	34
## 165	36	909	4	4	36
## 166	6	2978	1	2	32
## 167	18	1131	4	2	33
## 168	11	1577	4	1	20
## 170	24	1935	4	4	31
## 171	15	950	4	3	33
## 173	24	2064	3	2	34
## 174	8	1414	4	2	33
## 175	21	3414	2	1	26
## 177	12	2577	2	1	42
## 178	6	338	4	4	52
## 180	21	571	4	4	65
## 181	36	9572	1	1	28
## 184	24	3777	4	4	50
## 187	9	5129	2	4	74
## 190	18	3244	1	4	33
## 191	24	4591	2	3	54
## 194	6	2108	2	2	29
## 195	45	3031	4	4	21
## 197	6	1382	1	1	28
## 198	12	951	4	4	27
## 199	24	2760	4	4	36
## 200	18	4297	4	3	40
## 201	9	936	4	2	52
## 202	12	1168	4	3	27
## 203	27	5117	3	4	26
## 206	30	10623	3	4	38
## 207	12	1935	4	4	43

## 211	9	3074	1	2	33
## 214	30	1908	4	4	66
## 215	36	3342	4	2	51
## 216	6	932	1	3	39
## 219	24	3021	2	2	24
## 220	10	1364	2	4	64
## 221	12	625	4	1	26
## 223	12	707	4	2	30
## 225	15	4657	3	2	30
## 226	36	2613	4	2	27
## 229	9	1478	4	2	22
## 230	24	3149	4	1	22
## 231	36	4210	4	2	26
## 232	9	2507	2	4	51
## 233	12	2141	3	1	35
## 234	18	866	4	2	25
## 235	4	1544	2	1	42
## 238	21	2767	4	2	61
## 242	6	1595	3	2	51
## 244	12	1185	3	2	27
## 245	12	3447	4	3	35
## 247	12	717	4	4	52
## 253	30	2150	4	2	24
## 254	24	4151	2	3	35
## 255	9	2030	2	1	24
## 256	60	7418	1	1	27
## 259	15	3812	1	4	23
## 260	11	1154	4	4	57
## 261	12	1657	2	2	27
## 262	24	1603	4	4	55
## 263	18	5302	2	4	36
## 264	12	2748	2	4	57
## 265	10	1231	3	4	32
## 267	36	6304	4	4	36
## 269	14	8978	1	4	45
## 272	12	1402	3	4	37
## 273	48	12169	4	4	36
## 274	48	3060	4	4	28
## 276	9	2697	1	2	32
## 278	12	1262	2	4	49
## 279	6	4611	1	4	32
## 280	24	1901	4	4	29
## 286	47	10722	1	1	35
## 289	12	1092	4	4	49
## 290	24	1024	4	4	48
## 291	12	1076	2	2	26
## 292	36	9398	1	4	28
## 294	42	4796	4	4	56
## 295	48	7629	4	2	46
## 296	48	9960	1	2	26
## 297	12	4675	1	4	20
## 299	18	2515	3	4	43
## 300	21	2745	3	2	32
## 301	6	672	1	4	54

## 302	36	3804	4	1	42
## 306	6	1543	4	2	33
## 307	30	4811	2	4	24
## 308	12	727	4	3	33
## 315	4	1494	1	2	29
## 317	12	708	2	3	38
## 318	24	4351	1	4	48
## 319	12	701	4	2	32
## 322	24	1938	4	3	32
## 323	24	2910	2	1	34
## 327	12	5801	2	4	49
## 328	24	1525	4	3	34
## 330	6	1068	4	4	28
## 331	24	6615	2	4	75
## 332	18	1864	4	2	30
## 334	48	11590	2	4	24
## 335	24	4110	3	4	23
## 337	13	2101	2	4	23
## 338	15	1275	4	2	24
## 339	24	4169	4	4	28
## 344	18	4439	1	1	33
## 345	10	3949	1	1	37
## 347	13	882	4	4	23
## 348	24	3758	1	4	23
## 349	6	1743	1	2	34
## 351	9	1236	1	4	23
## 354	12	6199	4	2	28
## 355	10	727	4	4	46
## 360	30	2406	4	4	23
## 361	18	1239	4	4	61
## 362	12	3399	2	3	37
## 365	18	2473	4	1	25
## 366	12	1542	2	4	36
## 368	18	3650	1	4	22
## 370	18	3001	2	4	40
## 371	36	3079	4	4	36
## 372	18	6070	3	4	33
## 373	10	2146	1	3	23
## 374	60	13756	2	4	63
## 375	60	14782	3	4	60
## 377	18	2320	2	3	34
## 378	7	846	3	4	36
## 379	36	14318	4	2	57
## 385	30	4272	2	2	26
## 388	18	7374	4	4	40
## 389	15	2326	2	4	27
## 392	12	983	1	4	19
## 393	36	3249	2	4	39
## 394	6	1957	1	4	31
## 395	9	2406	2	3	31
## 397	12	2578	3	4	55
## 398	36	2348	3	2	46
## 399	12	1223	1	1	46
## 400	24	1516	4	1	43

## 401	18	1473	3	4	39
## 403	24	8648	2	2	27
## 405	18	2899	4	4	43
## 406	24	2039	1	1	22
## 407	24	2197	4	4	43
## 408	15	1053	4	2	27
## 409	24	3235	3	2	26
## 410	12	939	4	2	28
## 411	24	1967	4	4	20
## 413	12	2292	4	2	42
## 414	10	1597	3	2	40
## 415	24	1381	4	2	35
## 416	36	5842	2	2	35
## 417	12	2579	4	1	33
## 418	18	8471	1	2	23
## 419	21	2782	1	2	31
## 422	12	2028	4	2	30
## 423	12	958	2	3	47
## 424	21	1591	4	3	34
## 425	12	2762	1	2	25
## 430	18	1190	2	4	55
## 431	5	3448	1	4	74
## 435	9	2136	3	2	25
## 436	12	1484	2	1	25
## 439	42	3394	4	4	65
## 441	12	1884	4	4	39
## 445	48	5096	2	3	30
## 449	12	1424	3	4	55
## 450	15	1512	3	3	61
## 451	36	11054	4	2	30
## 452	6	518	3	1	29
## 453	12	2759	2	4	34
## 455	24	4817	2	3	31
## 456	24	2679	4	1	29
## 458	12	3386	3	4	35
## 460	18	4594	3	2	32
## 461	36	3620	1	2	37
## 464	12	754	4	4	38
## 466	24	2924	3	4	63
## 471	24	3092	3	2	22
## 472	6	448	4	4	23
## 473	9	654	4	3	28
## 475	18	1245	4	2	33
## 476	18	3114	1	4	26
## 477	39	2569	4	4	24
## 478	24	5152	4	2	25
## 480	15	1478	4	4	44
## 482	24	1201	4	1	26
## 484	15	960	3	2	30
## 485	12	1163	4	4	44
## 487	12	3077	2	4	52
## 489	10	1418	3	2	35
## 490	6	3518	2	3	26
## 494	6	368	4	4	38

## 495	12	2122	3	2	39
## 496	24	2996	2	4	20
## 498	24	1585	4	3	40
## 500	6	1323	2	4	28
## 504	24	1216	4	4	38
## 505	24	1207	4	4	24
## 507	15	2360	2	2	36
## 509	24	1413	4	2	28
## 510	39	8588	4	2	45
## 511	12	759	4	2	26
## 512	36	4686	2	2	32
## 513	15	2687	2	4	26
## 515	24	2255	4	1	54
## 516	6	609	4	3	37
## 517	6	1361	2	4	40
## 518	36	7127	2	4	23
## 519	6	1203	3	2	43
## 520	6	700	4	4	36
## 521	24	5507	3	4	44
## 522	18	3190	2	2	24
## 523	48	7119	3	4	53
## 524	24	3488	3	4	23
## 528	4	1503	2	1	42
## 529	36	2302	4	4	31
## 530	6	662	3	4	41
## 531	36	2273	3	1	32
## 532	15	2631	2	4	28
## 534	24	1311	4	3	26
## 536	21	2319	2	1	33
## 538	18	3612	3	4	37
## 541	12	1534	1	1	23
## 542	24	2032	4	4	60
## 544	18	2864	2	1	34
## 546	24	1333	4	2	43
## 547	24	2022	4	4	37
## 548	24	1552	3	1	32
## 550	48	8858	2	1	35
## 551	12	996	4	4	23
## 552	6	1750	2	4	45
## 553	48	6999	1	1	34
## 554	12	1995	4	1	27
## 558	21	5003	1	4	29
## 559	24	3552	3	4	27
## 565	24	4712	4	2	37
## 566	24	1553	3	2	23
## 567	12	1372	2	3	36
## 570	48	6758	3	2	31
## 571	24	3234	4	4	23
## 572	30	5954	3	2	38
## 574	15	806	4	4	22
## 575	9	1082	4	4	27
## 577	12	2930	2	1	27
## 580	24	937	4	3	27
## 584	36	2384	4	1	33

## 585	12	2133	4	4	52
## 587	9	2799	2	2	36
## 588	12	1289	4	1	21
## 589	18	1217	4	3	47
## 590	12	2246	3	3	60
## 596	6	931	1	1	32
## 598	24	4241	1	4	36
## 601	7	2329	1	1	45
## 602	9	918	4	1	30
## 605	10	1275	4	2	23
## 606	24	2828	4	4	22
## 607	24	4526	3	2	74
## 610	15	1300	4	4	45
## 611	12	741	4	3	22
## 617	60	9157	2	2	27
## 618	6	3676	1	3	37
## 619	30	3441	2	4	21
## 620	12	640	4	2	49
## 621	21	3652	2	3	27
## 624	12	1858	4	1	22
## 625	18	2600	4	4	65
## 626	15	1979	4	2	35
## 627	6	2116	2	2	41
## 628	9	1437	2	3	29
## 629	42	4042	4	4	36
## 630	9	3832	1	4	64
## 631	24	3660	2	4	28
## 632	18	1553	4	3	44
## 634	9	1980	2	2	19
## 636	12	1393	4	4	47
## 638	60	15653	2	4	21
## 639	12	1493	4	3	34
## 640	42	4370	3	2	26
## 642	15	1308	4	4	38
## 644	24	1851	4	2	33
## 646	36	7980	4	4	27
## 647	30	4583	2	2	32
## 649	24	947	4	3	38
## 650	12	684	4	4	40
## 652	12	1922	4	2	37
## 653	24	2303	4	1	45
## 654	36	8086	2	4	42
## 656	14	3973	1	4	22
## 658	48	10222	4	3	37
## 661	12	1297	3	4	23
## 662	12	900	4	2	23
## 663	21	2241	4	2	50
## 666	24	6314	4	2	27
## 667	30	3496	4	2	34
## 668	48	3609	1	1	27
## 669	12	4843	3	4	43
## 670	30	3017	4	4	47
## 671	24	4139	3	3	27
## 673	60	10366	2	4	42



## 674	6	2080	1	2	24
## 675	21	2580	4	2	41
## 676	30	4530	4	4	26
## 678	72	5595	2	2	24
## 682	12	2279	4	4	37
## 687	10	1347	4	2	27
## 688	36	2862	4	3	30
## 691	15	975	2	3	25
## 693	24	2896	2	1	29
## 694	6	4716	1	3	44
## 695	24	2284	4	2	28
## 696	6	1236	2	4	50
## 697	12	1103	4	3	29
## 699	18	1800	4	2	24
## 702	48	6331	4	4	46
## 703	24	1377	4	2	47
## 704	30	2503	4	2	41
## 706	15	5324	1	4	35
## 707	48	6560	3	2	24
## 709	9	1206	4	4	25
## 710	9	2118	2	2	37
## 711	18	629	4	3	32
## 712	6	1198	4	4	35
## 713	21	2476	4	4	46
## 718	18	1505	4	2	32
## 719	24	3148	3	2	31
## 720	20	6148	3	4	31
## 722	6	433	4	2	24
## 724	9	790	4	3	66
## 725	27	2570	3	3	21
## 727	15	1316	2	2	47
## 731	24	6403	1	2	33
## 733	8	760	4	2	44
## 734	24	2603	2	4	28
## 736	36	3990	3	2	29
## 738	18	4380	3	4	35
## 739	6	6761	1	3	45
## 741	24	2325	2	3	32
## 744	24	2483	4	4	22
## 748	12	1274	3	1	37
## 749	21	5248	1	3	26
## 751	6	428	2	1	49
## 752	18	976	1	2	23
## 753	12	841	2	4	23
## 754	30	5771	4	2	25
## 755	12	1555	4	4	55
## 756	24	1285	4	4	32
## 757	6	1299	1	1	74
## 758	15	1271	3	4	39
## 759	24	1393	2	2	31
## 760	12	691	4	3	35
## 761	15	5045	1	4	59
## 762	18	2124	4	4	24
## 766	12	1155	3	3	40

## 767	30	3108	2	4	31
## 768	10	2901	1	4	31
## 769	12	3617	1	4	28
## 771	24	2812	2	4	26
## 773	21	3275	1	4	36
## 775	12	1480	2	4	66
## 777	36	3535	4	4	37
## 779	36	5711	4	2	38
## 780	18	3872	2	4	67
## 781	39	4933	2	2	25
## 782	24	1940	4	4	60
## 783	12	1410	2	2	31
## 784	12	836	4	2	23
## 785	20	6468	1	4	60
## 786	18	1941	4	2	35
## 788	48	2751	4	3	38
## 789	48	6224	4	4	50
## 790	40	5998	4	3	27
## 791	21	1188	2	4	39
## 792	24	6313	3	4	41
## 793	6	1221	1	2	27
## 801	24	1597	4	4	54
## 805	12	7472	1	2	24
## 807	6	590	3	3	26
## 808	12	930	4	4	65
## 809	42	9283	1	2	55
## 810	15	1778	2	1	26
## 811	8	907	3	2	26
## 813	36	9629	4	4	24
## 814	48	3051	3	4	54
## 817	6	1338	1	4	62
## 818	6	1554	1	2	24
## 819	36	15857	2	3	43
## 825	18	3780	3	2	35
## 826	21	1602	4	3	30
## 827	18	3966	1	4	33
## 831	24	2375	4	2	44
## 833	45	11816	2	4	29
## 836	12	1082	4	4	48
## 838	4	601	1	3	23
## 843	18	1943	4	4	23
## 845	18	3422	4	4	47
## 847	18	6761	2	4	68
## 848	24	1249	4	2	28
## 849	9	1364	3	4	59
## 850	12	709	4	4	57
## 851	20	2235	4	2	33
## 854	18	1442	4	4	32
## 856	24	1474	4	3	33
## 857	10	894	4	3	40
## 858	15	3343	4	2	28
## 859	15	3959	3	2	29
## 860	9	3577	1	2	26
## 861	24	5804	4	2	27

## 862	18	2169	4	2	28
## 863	24	2439	4	4	35
## 864	27	4526	4	2	32
## 868	12	3331	2	4	42
## 869	36	7409	3	2	37
## 870	12	652	4	4	24
## 875	12	3590	2	2	29
## 877	18	1940	3	4	36
## 879	9	1422	3	2	27
## 880	30	6742	2	3	36
## 883	30	2181	4	4	36
## 884	18	1098	4	4	65
## 885	24	4057	3	3	43
## 886	12	795	4	4	53
## 887	24	2825	4	3	34
## 888	48	15672	2	2	23
## 889	36	6614	4	4	34
## 890	28	7824	3	4	40
## 891	27	2442	4	4	43
## 892	15	1829	4	4	46
## 893	12	2171	4	4	38
## 895	18	1169	4	3	29
## 896	36	8947	3	2	31
## 897	21	2606	4	4	28
## 898	12	1592	3	2	35
## 899	15	2186	1	4	33
## 900	18	4153	2	3	42
## 901	16	2625	2	4	43
## 902	20	3485	2	4	44
## 904	15	1386	4	2	40
## 905	24	1278	4	1	36
## 906	12	1107	2	2	20
## 907	21	3763	2	2	24
## 909	15	3594	1	2	46
## 910	9	3195	1	2	33
## 911	36	4454	4	4	34
## 912	24	4736	2	4	25
## 914	11	2142	1	2	28
## 918	6	14896	1	4	68
## 919	24	2359	1	1	33
## 922	48	12749	4	1	37
## 923	9	1366	3	4	22
## 924	12	2002	3	4	30
## 926	12	697	4	2	46
## 927	18	1049	4	4	21
## 930	12	1344	4	2	43
## 931	24	1747	4	1	24
## 934	12	522	4	4	42
## 935	12	1498	4	1	23
## 936	30	1919	4	3	30
## 937	9	745	3	2	28
## 939	60	6288	4	4	42
## 940	24	6842	2	4	46
## 942	10	1546	3	2	31

## 943	24	929	4	2	31
## 944	4	1455	2	1	42
## 948	12	2859	4	4	38
## 950	24	3621	2	4	31
## 953	24	4113	3	4	28
## 954	36	10974	4	2	26
## 960	24	3069	4	4	30
## 961	6	1740	2	2	30
## 962	21	2353	1	4	47
## 964	24	2397	3	2	35
## 966	30	1715	4	1	26
## 967	27	2520	4	2	23
## 968	15	3568	4	2	54
## 970	11	3939	1	2	40
## 972	24	7393	1	4	43
## 974	60	7297	4	4	36
## 975	30	2831	4	2	33
## 976	24	1258	3	3	57
## 977	6	753	2	3	64
## 978	18	2427	4	2	42
## 979	24	2538	4	4	47
## 980	15	1264	2	2	25
## 981	30	8386	2	2	49
## 982	48	4844	3	2	33
## 984	36	8229	2	2	26
## 985	24	2028	2	2	30
## 986	15	1433	4	3	25
## 988	13	1409	2	4	64
## 989	24	6579	4	2	29
## 990	24	1743	4	2	48
## 991	12	3565	2	1	37
## 992	15	1569	4	4	34
## 994	36	3959	4	3	30
## 995	12	2390	4	3	50
## 997	30	3857	4	4	40
## 998	12	804	4	4	38
## 999	45	1845	4	4	23
##	NumberExistingCredits	NumberPeopleMaintenance	Telephone	ForeignWorker	Class
## 2	1	1	1	1	Bad
## 3	1	2	1	1	Good
## 4	1	2	1	1	Good
## 5	2	2	1	1	Bad
## 6	1	2	0	1	Good
## 7	1	1	1	1	Good
## 9	1	1	1	1	Good
## 10	2	1	1	1	Bad
## 11	1	1	1	1	Bad
## 12	1	1	1	1	Bad
## 13	1	1	0	1	Good
## 14	2	1	1	1	Bad
## 15	1	1	1	1	Good
## 16	1	1	1	1	Bad
## 17	2	1	1	1	Good
## 21	3	1	0	1	Good

## 22	1	2	1	1	Good
## 23	2	2	1	0	Good
## 24	1	1	1	1	Good
## 27	1	1	1	1	Good
## 31	1	1	0	1	Good
## 32	1	1	1	1	Good
## 33	2	1	0	1	Good
## 34	1	1	1	1	Good
## 36	2	1	1	1	Bad
## 37	1	1	0	1	Good
## 38	1	1	1	1	Bad
## 40	1	1	1	1	Good
## 42	1	1	0	1	Good
## 43	1	2	0	1	Good
## 45	2	1	1	1	Bad
## 50	1	1	1	1	Good
## 51	1	1	1	1	Good
## 52	2	1	0	1	Good
## 55	2	1	0	1	Bad
## 58	2	1	1	1	Good
## 59	1	1	1	1	Good
## 60	2	1	0	1	Bad
## 66	4	2	0	1	Good
## 68	1	1	1	1	Good
## 69	1	1	0	1	Bad
## 70	1	1	1	1	Good
## 71	1	1	1	1	Good
## 72	2	1	0	1	Good
## 77	1	1	1	1	Bad
## 78	1	1	1	1	Good
## 81	2	1	0	1	Bad
## 83	1	1	1	1	Good
## 84	1	1	0	1	Good
## 85	1	1	1	1	Good
## 86	2	1	0	1	Good
## 88	1	2	0	1	Bad
## 90	2	1	1	1	Bad
## 91	1	1	1	1	Good
## 92	1	1	1	1	Good
## 93	1	2	1	1	Bad
## 96	1	1	0	1	Bad
## 97	1	1	1	1	Good
## 98	1	1	1	1	Good
## 99	1	1	1	1	Good
## 100	2	2	0	1	Good
## 101	1	1	1	1	Good
## 102	1	1	1	1	Good
## 103	1	1	1	1	Good
## 104	1	1	0	1	Good
## 106	2	2	0	1	Bad
## 109	1	1	0	0	Good
## 110	1	1	0	1	Good
## 111	2	2	1	1	Good
## 115	1	1	1	1	Good

## 117	1	1	0	1	Bad
## 118	2	1	1	0	Good
## 119	2	1	1	1	Bad
## 120	1	1	0	1	Good
## 121	2	1	0	1	Bad
## 123	1	1	1	1	Good
## 125	1	1	1	1	Bad
## 126	2	1	1	1	Good
## 127	1	1	1	1	Good
## 128	1	1	1	1	Bad
## 129	2	1	0	1	Good
## 130	2	1	1	1	Bad
## 131	1	1	1	1	Good
## 134	2	1	1	1	Good
## 135	1	1	0	1	Good
## 136	2	1	0	1	Good
## 137	2	1	1	1	Good
## 139	3	1	0	1	Good
## 141	1	1	1	0	Good
## 142	1	1	0	1	Good
## 145	1	1	0	1	Good
## 146	1	1	1	1	Good
## 147	2	1	0	1	Good
## 150	2	1	1	1	Good
## 151	1	2	0	1	Good
## 153	1	1	1	1	Good
## 154	1	1	1	1	Good
## 155	1	1	0	1	Good
## 156	1	1	1	1	Bad
## 157	2	2	1	0	Good
## 158	1	1	1	1	Good
## 159	2	1	0	1	Good
## 160	2	2	1	1	Good
## 165	1	1	1	1	Good
## 166	1	1	0	1	Good
## 167	1	1	1	1	Bad
## 168	1	1	1	1	Good
## 170	2	1	0	1	Bad
## 171	2	2	1	1	Bad
## 173	1	1	0	1	Bad
## 174	1	1	1	0	Good
## 175	2	1	1	1	Bad
## 177	1	1	1	1	Good
## 178	2	1	1	1	Good
## 180	2	1	1	1	Good
## 181	2	1	1	1	Bad
## 184	1	1	0	1	Good
## 187	1	2	0	1	Bad
## 190	2	1	0	1	Good
## 191	3	1	0	1	Bad
## 194	1	1	1	1	Good
## 195	1	1	1	1	Bad
## 197	2	1	0	1	Good
## 198	4	1	1	1	Bad

## 199	1	1	0	1	Good
## 200	1	1	0	1	Bad
## 201	2	1	0	1	Good
## 202	1	1	1	1	Good
## 203	2	1	1	1	Good
## 206	3	2	0	1	Good
## 207	3	1	0	1	Good
## 211	2	2	1	1	Good
## 214	1	1	0	1	Bad
## 215	1	1	0	1	Good
## 216	2	1	1	1	Good
## 219	1	1	1	1	Good
## 220	1	1	0	1	Good
## 221	1	1	1	1	Good
## 223	2	1	1	1	Good
## 225	1	1	0	1	Good
## 226	2	1	1	1	Good
## 229	1	1	1	1	Bad
## 230	1	1	1	1	Good
## 231	1	1	1	1	Bad
## 232	1	1	1	1	Good
## 233	1	1	1	1	Good
## 234	1	1	1	1	Good
## 235	3	2	1	1	Good
## 238	2	1	1	1	Bad
## 242	1	2	1	1	Good
## 244	2	1	1	1	Good
## 245	1	2	1	1	Good
## 247	3	1	1	1	Good
## 253	1	1	1	1	Bad
## 254	2	1	1	1	Good
## 255	1	1	0	1	Good
## 256	1	1	1	1	Good
## 259	1	1	0	1	Good
## 260	3	1	1	1	Good
## 261	1	1	1	1	Good
## 262	1	1	1	1	Good
## 263	3	1	0	1	Good
## 264	3	1	1	1	Good
## 265	2	2	1	0	Good
## 267	2	1	1	1	Good
## 269	1	1	0	0	Bad
## 272	1	1	0	1	Good
## 273	1	1	0	1	Good
## 274	2	1	1	1	Bad
## 276	1	2	1	1	Good
## 278	1	1	0	1	Good
## 279	1	1	1	1	Bad
## 280	1	1	0	1	Good
## 286	1	1	0	1	Good
## 289	2	1	0	1	Good
## 290	1	1	1	1	Bad
## 291	1	1	0	0	Good
## 292	1	1	0	1	Bad

## 294	1	1	1	1	Good
## 295	2	2	1	1	Good
## 296	1	1	0	1	Bad
## 297	1	1	1	1	Good
## 299	1	1	0	1	Good
## 300	2	1	0	1	Good
## 301	1	1	0	1	Good
## 302	1	1	0	1	Bad
## 306	1	1	1	1	Good
## 307	1	1	1	1	Good
## 308	1	1	0	1	Bad
## 315	1	2	1	0	Good
## 317	1	2	1	1	Good
## 318	1	1	0	1	Good
## 319	2	1	1	1	Good
## 322	1	1	1	1	Bad
## 323	1	1	0	1	Good
## 327	1	1	0	1	Good
## 328	1	2	0	1	Good
## 330	1	2	1	1	Good
## 331	2	1	0	1	Good
## 332	2	1	1	1	Bad
## 334	2	1	1	1	Bad
## 335	2	2	1	1	Bad
## 337	1	1	1	1	Good
## 338	1	1	1	1	Bad
## 339	1	1	1	1	Good
## 344	1	1	0	1	Good
## 345	1	2	1	1	Good
## 347	2	1	1	1	Good
## 348	1	1	1	1	Good
## 349	2	1	1	1	Good
## 351	1	1	0	1	Good
## 354	2	1	0	1	Bad
## 355	1	1	0	1	Good
## 360	1	1	1	1	Bad
## 361	1	1	1	1	Good
## 362	1	1	1	1	Good
## 365	1	1	1	1	Bad
## 366	1	1	0	1	Good
## 368	1	1	1	1	Good
## 370	1	1	1	1	Good
## 371	1	1	1	1	Good
## 372	2	1	0	1	Good
## 373	2	1	1	1	Good
## 374	1	1	0	1	Good
## 375	2	1	0	1	Bad
## 377	2	1	1	1	Good
## 378	1	1	1	1	Good
## 379	1	1	0	1	Bad
## 385	2	1	1	1	Good
## 388	2	1	0	1	Good
## 389	1	1	1	1	Good
## 392	1	1	1	1	Good



## 393	1	2	0	1	Good
## 394	1	1	1	1	Good
## 395	1	1	1	1	Good
## 397	1	1	1	1	Good
## 398	2	1	0	1	Good
## 399	2	1	1	1	Bad
## 400	2	1	1	1	Good
## 401	1	1	0	1	Good
## 403	2	1	0	1	Bad
## 405	1	2	1	1	Good
## 406	1	1	0	1	Bad
## 407	2	2	0	1	Good
## 408	1	1	1	0	Good
## 409	1	1	0	1	Good
## 410	3	1	0	1	Bad
## 411	1	1	0	1	Good
## 413	2	1	0	1	Bad
## 414	1	2	1	0	Good
## 415	1	1	1	1	Bad
## 416	2	2	0	1	Good
## 417	1	2	1	1	Bad
## 418	2	1	0	1	Good
## 419	1	1	1	1	Good
## 422	1	1	1	1	Good
## 423	2	2	1	1	Good
## 424	2	1	1	1	Good
## 425	1	1	0	1	Bad
## 430	3	2	1	1	Bad
## 431	1	1	1	1	Good
## 435	1	1	1	1	Good
## 436	1	1	0	1	Bad
## 439	2	1	1	1	Good
## 441	1	1	0	1	Good
## 445	1	1	0	1	Bad
## 449	1	1	0	1	Good
## 450	2	1	1	1	Bad
## 451	1	1	0	1	Good
## 452	1	1	1	1	Good
## 453	2	1	1	1	Good
## 455	1	1	0	1	Bad
## 456	1	1	0	1	Good
## 458	1	1	0	1	Bad
## 460	1	1	0	1	Good
## 461	1	2	1	1	Good
## 464	2	1	1	1	Good
## 466	1	2	0	1	Good
## 471	1	1	0	1	Bad
## 472	1	1	1	1	Bad
## 473	1	1	1	1	Bad
## 475	1	1	1	1	Bad
## 476	1	1	1	1	Bad
## 477	1	1	1	1	Good
## 478	1	1	1	1	Good
## 480	2	2	0	1	Good

## 482	1	1	1	1	Good
## 484	2	1	1	1	Good
## 485	1	1	0	1	Good
## 487	1	1	0	1	Good
## 489	1	1	1	0	Good
## 490	1	1	1	1	Good
## 494	1	1	1	1	Good
## 495	2	2	1	0	Good
## 496	1	1	1	1	Bad
## 498	2	1	1	1	Good
## 500	2	2	0	1	Good
## 504	2	2	1	1	Bad
## 505	1	1	1	1	Bad
## 507	1	1	0	1	Good
## 509	1	1	1	1	Good
## 510	1	1	0	1	Good
## 511	1	1	1	1	Bad
## 512	1	1	0	1	Good
## 513	1	1	0	1	Good
## 515	1	1	1	1	Good
## 516	2	1	1	0	Good
## 517	1	2	1	0	Good
## 518	2	1	0	1	Bad
## 519	1	1	0	1	Good
## 520	2	1	1	1	Good
## 521	2	1	1	1	Good
## 522	1	1	1	1	Bad
## 523	2	2	1	1	Bad
## 524	1	1	1	1	Good
## 528	2	2	1	1	Good
## 529	1	1	1	1	Bad
## 530	1	2	0	1	Good
## 531	2	2	1	1	Good
## 532	2	1	0	1	Bad
## 534	1	1	0	1	Good
## 536	1	1	1	1	Bad
## 538	1	1	0	1	Good
## 541	1	1	1	1	Bad
## 542	2	1	0	1	Good
## 544	1	2	1	1	Bad
## 546	2	2	1	1	Bad
## 547	1	1	0	1	Good
## 548	1	2	1	1	Good
## 550	2	1	0	1	Good
## 551	2	1	1	1	Good
## 552	1	2	1	1	Good
## 553	2	1	0	1	Bad
## 554	1	1	1	1	Good
## 558	2	1	0	1	Bad
## 559	1	1	1	1	Bad
## 565	2	1	0	1	Good
## 566	2	1	0	1	Good
## 567	1	1	1	1	Bad
## 570	1	1	0	1	Bad

## 571	1	1	0	1	Bad
## 572	1	1	1	1	Good
## 574	1	1	1	1	Good
## 575	2	1	1	1	Good
## 577	1	1	1	1	Good
## 580	2	1	1	1	Good
## 584	1	1	1	1	Bad
## 585	1	1	0	1	Good
## 587	2	2	1	1	Good
## 588	1	1	1	1	Good
## 589	1	1	0	1	Bad
## 590	2	1	1	1	Bad
## 596	1	1	1	1	Bad
## 598	3	1	0	1	Bad
## 601	1	1	1	1	Good
## 602	1	1	1	1	Bad
## 605	1	1	1	1	Good
## 606	1	1	0	1	Good
## 607	1	1	0	1	Good
## 610	1	2	1	1	Good
## 611	1	1	1	1	Bad
## 617	1	1	1	1	Good
## 618	3	2	1	1	Good
## 619	1	1	1	1	Bad
## 620	1	1	1	1	Good
## 621	2	1	1	1	Good
## 624	1	1	1	1	Good
## 625	2	1	1	1	Bad
## 626	1	1	1	1	Good
## 627	1	1	0	1	Good
## 628	1	1	1	1	Bad
## 629	2	1	0	1	Good
## 630	1	1	1	1	Good
## 631	1	1	1	1	Good
## 632	1	1	1	1	Bad
## 634	2	1	1	1	Bad
## 636	3	2	0	1	Good
## 638	2	1	0	1	Good
## 639	1	2	1	1	Good
## 640	2	2	0	1	Bad
## 642	2	1	1	1	Good
## 644	2	1	0	1	Good
## 646	2	1	0	1	Bad
## 647	2	1	1	1	Good
## 649	1	2	1	1	Bad
## 650	1	2	1	1	Bad
## 652	1	1	1	1	Bad
## 653	1	1	1	1	Bad
## 654	4	1	0	1	Bad
## 656	1	1	1	1	Good
## 658	1	1	0	1	Good
## 661	1	1	1	1	Good
## 662	1	1	1	1	Bad
## 663	2	1	1	1	Good

## 666	2	1	0	1	Good
## 667	1	2	0	1	Good
## 668	1	1	1	1	Good
## 669	2	1	0	1	Bad
## 670	1	1	1	1	Good
## 671	2	1	0	1	Good
## 673	1	1	0	1	Good
## 674	1	1	1	1	Good
## 675	1	2	1	1	Bad
## 676	1	1	0	1	Good
## 678	1	1	1	1	Bad
## 682	1	1	0	1	Good
## 687	2	1	0	1	Good
## 688	1	1	1	1	Good
## 691	2	1	1	1	Good
## 693	1	1	1	1	Good
## 694	2	2	1	1	Good
## 695	1	1	0	1	Good
## 696	1	1	1	1	Good
## 697	2	1	1	0	Good
## 699	2	1	1	1	Good
## 702	2	1	0	1	Bad
## 703	1	1	0	1	Good
## 704	2	1	1	1	Good
## 706	1	1	1	1	Good
## 707	1	1	1	1	Bad
## 709	1	1	1	1	Good
## 710	1	2	1	1	Good
## 711	2	1	0	1	Good
## 712	1	1	1	1	Bad
## 713	1	1	0	1	Good
## 718	1	1	0	1	Good
## 719	2	1	0	1	Good
## 720	2	1	0	1	Good
## 722	1	2	1	1	Bad
## 724	1	1	1	1	Good
## 725	1	1	1	1	Bad
## 727	2	1	1	1	Good
## 731	1	1	1	1	Good
## 733	1	1	1	1	Good
## 734	1	1	0	1	Good
## 736	1	1	1	1	Good
## 738	1	2	0	1	Good
## 739	2	2	0	1	Good
## 741	1	1	1	1	Good
## 744	1	1	0	1	Good
## 748	1	1	1	1	Bad
## 749	1	1	1	1	Good
## 751	1	1	0	1	Good
## 752	1	1	1	1	Bad
## 753	1	1	1	1	Good
## 754	2	1	1	1	Good
## 755	2	2	1	1	Bad
## 756	1	1	1	1	Bad

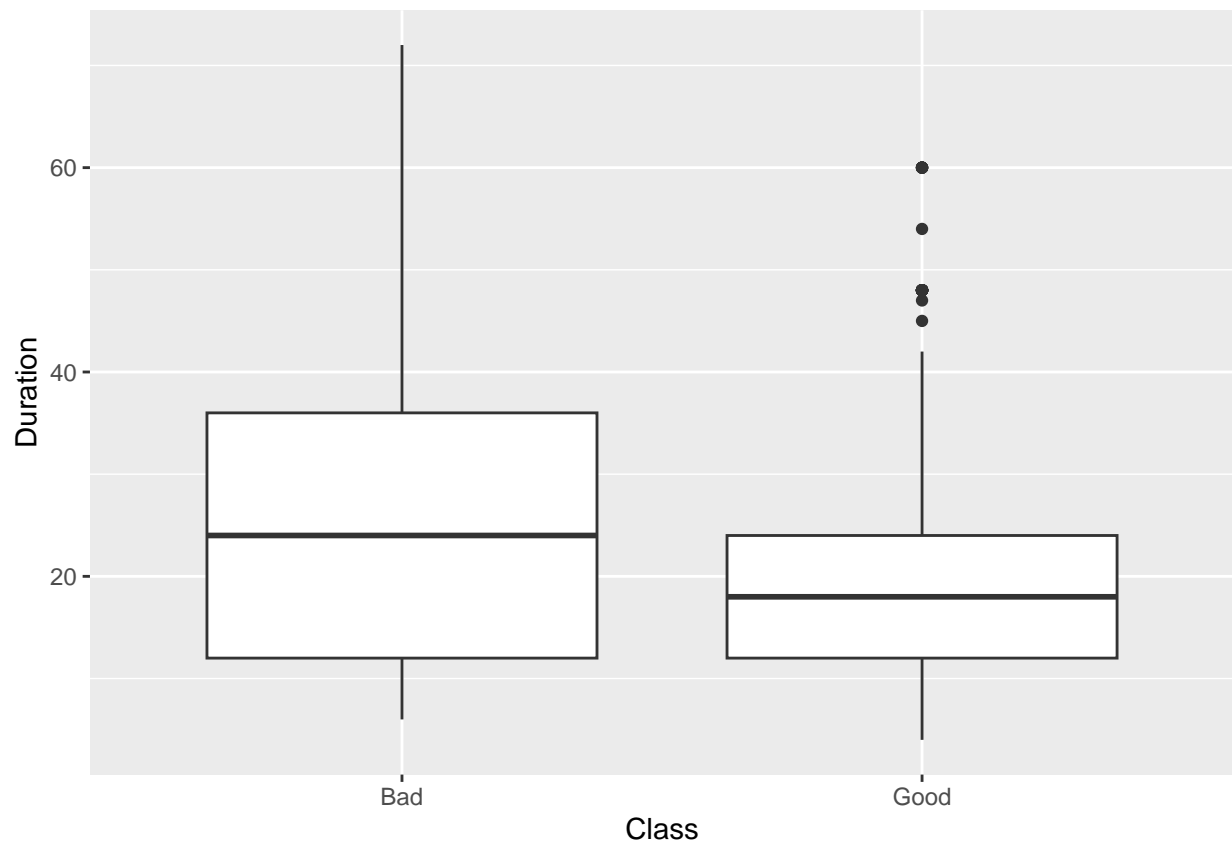
## 757	3	2	1	0	Good
## 758	2	1	0	1	Bad
## 759	1	1	0	1	Good
## 760	2	1	1	1	Bad
## 761	1	1	0	1	Good
## 762	2	1	1	1	Bad
## 766	2	1	1	1	Good
## 767	1	1	1	1	Bad
## 768	1	1	1	1	Good
## 769	3	1	0	1	Good
## 771	1	1	1	1	Good
## 773	1	1	0	1	Good
## 775	3	1	1	1	Good
## 777	2	1	0	1	Good
## 779	2	1	0	1	Good
## 780	1	1	0	1	Good
## 781	2	1	1	1	Bad
## 782	1	1	0	1	Good
## 783	1	1	0	1	Good
## 784	1	1	1	1	Bad
## 785	1	1	0	1	Good
## 786	1	1	0	1	Good
## 788	2	2	0	1	Good
## 789	1	1	1	1	Bad
## 790	1	1	0	1	Bad
## 791	1	2	1	1	Bad
## 792	1	2	0	1	Good
## 793	2	1	1	1	Good
## 801	2	2	1	1	Good
## 805	1	1	1	1	Good
## 807	1	1	1	0	Good
## 808	4	1	1	1	Good
## 809	1	1	0	1	Good
## 810	2	1	1	1	Bad
## 811	1	1	0	1	Good
## 813	2	1	0	1	Bad
## 814	1	1	1	1	Bad
## 817	1	1	1	1	Good
## 818	2	1	0	1	Good
## 819	1	1	1	1	Good
## 825	2	1	0	1	Good
## 826	2	1	0	1	Good
## 827	3	1	0	1	Bad
## 831	2	2	0	1	Good
## 833	2	1	1	1	Bad
## 836	2	1	1	1	Bad
## 838	1	2	1	1	Good
## 843	1	1	1	1	Bad
## 845	3	2	0	1	Good
## 847	2	1	1	1	Bad
## 848	1	1	1	1	Good
## 849	1	1	1	1	Good
## 850	1	1	1	1	Bad
## 851	2	1	1	0	Bad

## 854	2	2	1	1	Bad
## 856	1	1	0	1	Good
## 857	1	1	0	1	Good
## 858	1	1	0	1	Good
## 859	1	1	0	1	Bad
## 860	1	2	1	0	Good
## 861	2	1	1	1	Good
## 862	1	1	0	1	Bad
## 863	1	1	0	1	Bad
## 864	2	2	0	1	Good
## 868	1	1	1	1	Good
## 869	2	1	1	1	Good
## 870	1	1	1	1	Good
## 875	1	2	1	1	Good
## 877	1	1	0	1	Good
## 879	1	1	0	1	Bad
## 880	2	1	1	1	Good
## 883	2	1	1	1	Good
## 884	2	1	1	1	Good
## 885	1	1	0	1	Bad
## 886	1	1	1	1	Bad
## 887	2	2	0	1	Good
## 888	1	1	0	1	Bad
## 889	2	1	0	1	Good
## 890	2	2	0	1	Good
## 891	4	2	0	1	Good
## 892	2	1	0	1	Good
## 893	2	1	1	0	Good
## 895	2	1	0	1	Good
## 896	1	2	0	1	Good
## 897	1	1	0	1	Good
## 898	1	1	1	0	Good
## 899	1	1	1	1	Good
## 900	1	1	1	1	Bad
## 901	1	1	0	1	Bad
## 902	2	1	0	1	Good
## 904	1	1	0	1	Good
## 905	1	1	0	1	Good
## 906	1	2	0	1	Good
## 907	1	1	1	0	Good
## 909	2	1	1	1	Good
## 910	1	1	1	1	Good
## 911	2	1	1	1	Good
## 912	1	1	1	1	Bad
## 914	1	1	0	1	Good
## 918	1	1	0	1	Bad
## 919	1	1	1	1	Bad
## 922	1	1	0	1	Good
## 923	1	1	1	1	Bad
## 924	1	2	0	1	Good
## 926	2	1	0	1	Bad
## 927	1	1	1	1	Good
## 930	2	2	1	1	Good
## 931	1	1	1	0	Good

## 934	2	2	0	1	Good
## 935	1	1	1	1	Good
## 936	2	1	1	1	Bad
## 937	1	1	1	1	Bad
## 939	1	1	1	1	Bad
## 940	2	2	0	1	Good
## 942	1	2	1	0	Good
## 943	1	1	0	1	Good
## 944	3	2	1	1	Good
## 948	1	1	0	1	Good
## 950	2	1	1	1	Bad
## 953	1	1	1	1	Bad
## 954	2	1	0	1	Bad
## 960	1	1	1	1	Good
## 961	2	1	1	1	Good
## 962	2	1	1	1	Good
## 964	2	1	0	1	Bad
## 966	1	1	1	1	Good
## 967	2	1	1	1	Bad
## 968	1	1	0	1	Good
## 970	2	2	1	1	Good
## 972	1	2	1	1	Good
## 974	1	1	1	1	Bad
## 975	1	1	0	1	Good
## 976	1	1	1	1	Good
## 977	1	1	1	1	Good
## 978	2	1	1	1	Good
## 979	2	2	1	1	Bad
## 980	1	1	1	1	Bad
## 981	1	1	1	1	Bad
## 982	1	1	0	1	Bad
## 984	1	2	1	1	Bad
## 985	2	1	1	1	Good
## 986	2	1	1	1	Good
## 988	1	1	1	1	Good
## 989	1	1	0	1	Good
## 990	2	1	1	1	Good
## 991	2	2	1	1	Good
## 992	1	2	1	1	Good
## 994	1	1	0	1	Good
## 995	1	1	0	1	Good
## 997	1	1	0	1	Good
## 998	1	1	1	1	Good
## 999	1	1	0	1	Bad

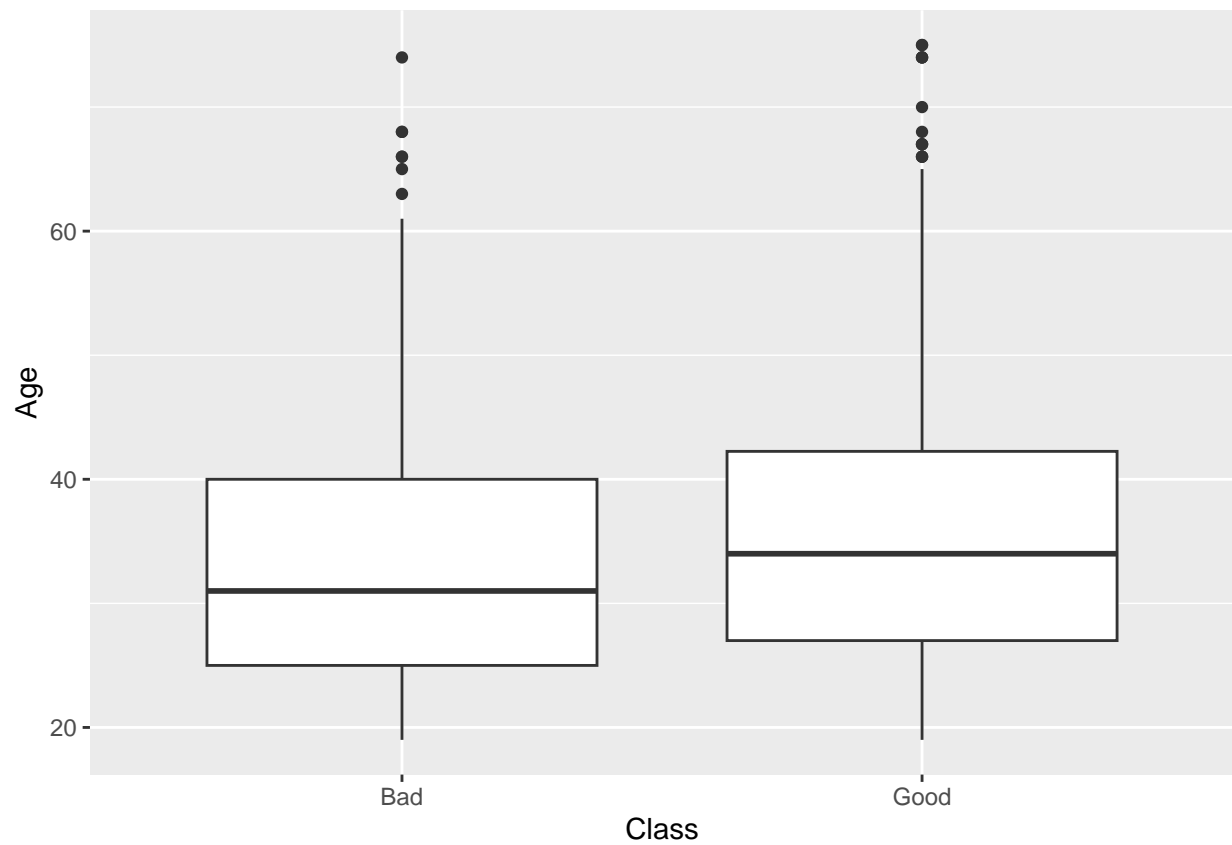
7. Create and interpret boxplots of all the predictor variables in relation to the outcome variable (**Class**).

```
ggplot(subCredit, aes(Class,Duration)) + geom_boxplot()
```

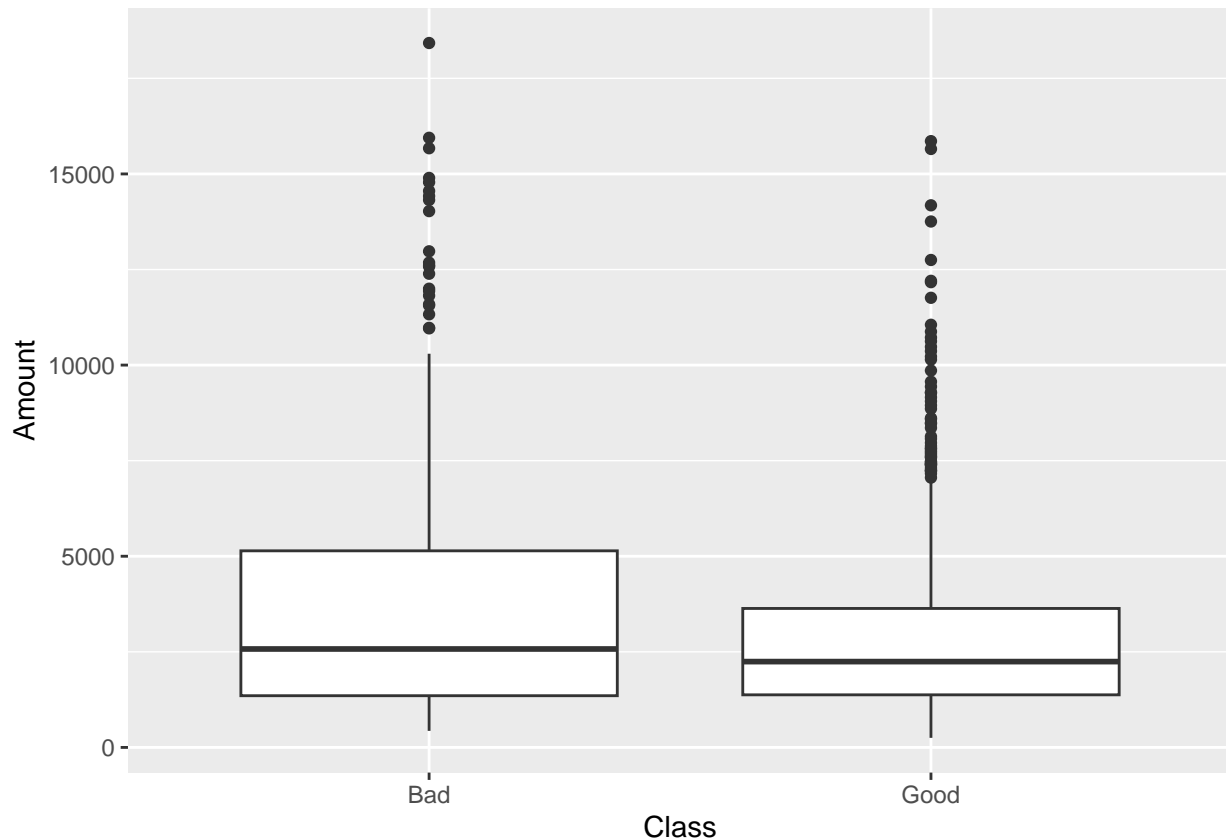


```
ggplot(subCredit, aes(Class, Age)) + geom_boxplot()
```





```
ggplot(subCredit, aes(Class,Amount)) + geom_boxplot()
```



8. Train a support vector machine with the `ksvm()` function from the `kernlab` package. Make sure that you have installed and libaried the `kernlab` package. Have the `cost` be 5, and have `ksvm` do 3 **cross validations** (Hint: try `prob.model = TRUE`)

```
svm <- ksvm(data=subCredit, Class ~.,C=5, CV=3,prob.model=TRUE)
svm
```

```
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc (classification)
## parameter : cost C = 5
##
## Gaussian Radial Basis kernel function.
## Hyperparameter : sigma = 0.092244004453622
##
## Number of Support Vectors : 634
##
## Objective Function Value : -2483.01
## Training error : 0.224
## Probability model included.
```

9. Examine the `ksvm` output object. In particular, look at the **cross-validation error** for an initial indication of model quality. Add a comment that gives your opinion on whether this is a good model.

```
# 0.219 is the error, and it is good because it is a low number
```

10. Predict the training cases using the `predict()` command

```
predicted <- predict(svm, testSet)
predicted
```

```
## [1] Good Good Good Bad Good Good Good Good Good Bad Good Good Good Good Good
## [16] Good Good Good Good Good Good Good Good Good Bad Good Good Good Good Good
## [31] Bad Good Good Good Good Good Good Bad Good Good Good Good Good Good Good
## [46] Good Bad Good Good Good Good Bad Good Good Good Bad Bad Good Good Good
## [61] Good Good Bad Good Good Bad Good Good Good Good Good Bad Good Good Good
## [76] Good Good Good Good Good Good Good Good Good Good Good Good Good Good
## [91] Good Bad Good Good Good Good Good Good Good Good Good Good Good Good
## [106] Good Good Good Bad Good Good Good Good Good Good Good Good Good Good Bad Good
## [121] Good Bad Good Bad Good Good Good Good Good Good Good Good Good Good Good
## [136] Good Good Good Good Good Good Good Good Good Good Good Good Good Good
## [151] Good Good Good Good Good Good Good Good Good Good Good Good Good Good
## [166] Bad Good Good Bad Good Good Good Good Good Good Good Good Good Good Bad Good
## [181] Bad Good Good Good Good Good Good Good Good Good Good Good Good Good
## [196] Good Good Good Good Good Bad Bad Good Good Good Good Good Good Good
## [211] Good Good Good Good Good Good Good Good Good Good Good Good Good Good Bad
## [226] Good Good Bad Good Good Good Good Good Good Good Good Good Good Good
## [241] Bad Good Good Good Good Good Good Good Good Good Good Good Good Bad Good
## [256] Good Good Good Good Bad Good Good Good Bad Good Good Good Good Bad Good
## [271] Good Good Good Good Good Good Good Good Good Good Good Good Good Bad Good
## [286] Good Good Good Good Good Good Good Good Good Good Good Good Good Bad Good
## [301] Good Good Good Good Good Good Good Good Bad Good Good Good Good Bad Good
## [316] Good Good Good Good Good Good Good Good Good Good Good Good Bad Good Bad
## [331] Good Good Bad Good Good Good Good Good Good Good Bad Good Good Good Good
## [346] Good Good Good Good Good Good Good Good Bad Good Good Good Good Good Bad
## [361] Good Bad Good Good Good Good Good Good Good Good Good Good Good Good Good
## [376] Good Good Good Bad Good Good Good Bad Good Bad Bad Good Good Bad Good
## [391] Good Good Good Good Good Good Good Good Good Good Good Good Good Bad Good
## [406] Good Good Good Good Good Good Good Good Good Good Good Good Good Good Bad
## [421] Good Good Good Good Good Good Good Good Bad Good Good Good Good Good
## [436] Good Good Good Good Good Good Good Good Good Good Good Good Bad Good Good
## [451] Good Good Good Good Good Good Good Good Good Good Good Good Good Good
## [466] Good Good Good Good Good Good Bad Bad Good Good Good Bad Good Good Good
## [481] Good Good Good Bad Bad Good Good Good Good Good Good Good Good Bad Good
## [496] Good Good Good Good Good Good Good Bad Good Good Good Good Good Good Good
## [511] Good Good Good Good Good Good Good Good Good Good Good Good Good Good Bad
## [526] Good Good Good Good Good Good Good Good Good Good Good Good Good Good
## [541] Good Good Good Good Good Good Good Bad Good Good Good Good Good Good Bad
## [556] Good Good Good Good Good Bad Good Good Good Good Good Good Good Bad Good
## [571] Good Good Good Good Good Good Good Good Bad Good Good Good Good Bad Good
## [586] Good Bad Good Good Good Good Good Good Good Good Bad Good Good Good Bad
## Levels: Bad Good
```

- Examine the predicted out object with `str( )`. Then, calculate a **confusion matrix** using the `table()` function.

```
str(predicted)
```

```
## Factor w/ 2 levels "Bad","Good": 2 2 2 1 2 2 2 2 2 1 ...
```

```
CCM <- table(predicted, testSet$Class)
```

```
CCM
```

```
##
```

```
## predicted Bad Good
```

```
##      Bad   56   14
```

```
##      Good 124  406
```

12. Interpret the confusion matrix and in particular calculate the overall **accuracy** of the model. The **diag( )** command can be applied to the results of the table command you ran in the previous step. You can also use **sum( )** to get the total of all four cells.

```
(sum(CCM)- sum(diag(CCM)))/sum(CCM)
```

```
## [1] 0.23
```

13. Check you calculation with the **confusionMatrix()** function in the **caret** package.

```
1-0.2183333
```

```
## [1] 0.7816667
```