

Amin Baabol
Partner: Mohamed Ahmed
INFS 762
Project 1
September 30th, 2020

Task 1: Classification (with TARGET_B as the dependent variable)

Step 1

1.1: Importing

1.2: Code

```
9  /**Step 1**/  
10 /*1.2 Dropping variables*/  
11  
12 DATA WORK.kddcup98;  
13 SET WORK.kddcup98;  
14 drop ID Var29 Var30;  
15 RUN;  
16  
17
```

Step 2: Data exploration

2.1: Histograms

Variable list:

- DemAge
- DemMedHomeValue
- DemMedIncome
- DemPctVeterans
- GiftAvg36
- GiftAvgAll
- GiftAvgCard36
- GiftAvgLast
- GiftCnt36
- GiftCntAll
- GiftCntCard36
- GiftCntCardAll
- GiftTimeFirst
- GiftTimeLast
- PromCnt12
- PromCnt36
- PromCntAll
- PromCntCard12
- PromCntCard36
- PromCntCardAll

Code

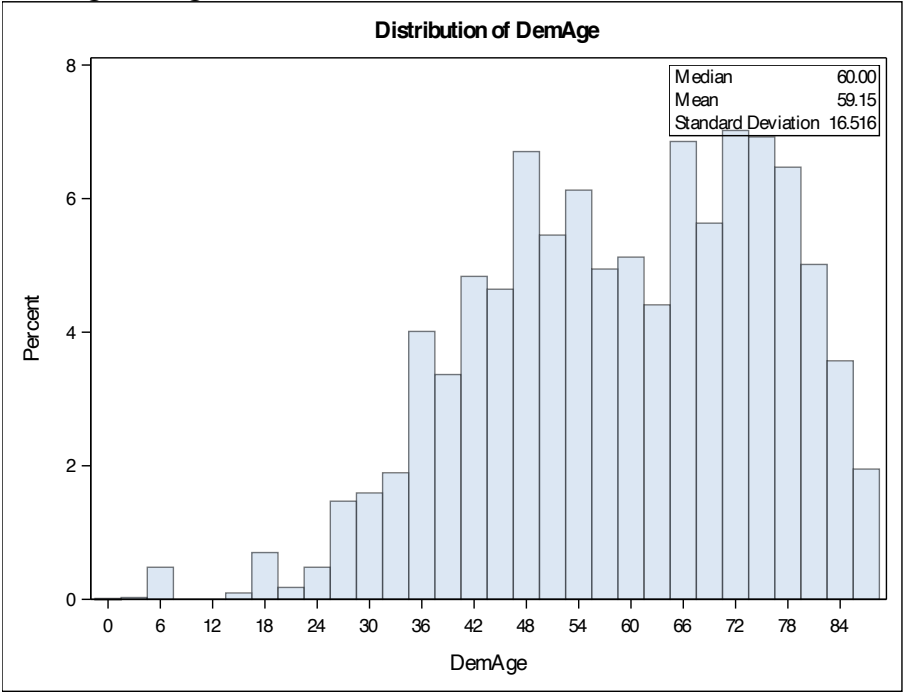
```
19 /** Step 2**/  
20 /*2.1 Histograms*/  
21 proc univariate data=WORK.kddcup98 noprint;  
22     histogram DemAge;  
23     title 'histogram for DemAge';  
24     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)  
25     / POSITION = ne;  
26 run;  
27 proc univariate data=WORK.kddcup98 noprint;  
28     histogram DemMedHomeValue;  
29     title 'histogram for DemMedHomeValue';  
30     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)  
31     / POSITION = ne;  
32 run;  
33 proc univariate data=WORK.kddcup98 noprint;  
34     histogram DemMedIncome;  
35     title 'histogram for DemMedIncome';  
36     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)  
37     / POSITION = ne;  
38 run;  
39 proc univariate data=WORK.kddcup98 noprint;  
40     histogram DemPctVeterans;  
41     title 'histogram for DemPctVeterans';  
42     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)  
43     / POSITION = ne;  
44 run;  
45 proc univariate data=WORK.kddcup98 noprint;  
46     histogram GiftAvg36;  
47     title 'histogram for GiftAvg36';  
48     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)  
49     / POSITION = ne;  
50 run;  
51 proc univariate data=WORK.kddcup98 noprint;  
52     histogram GiftAvgAll;  
53     title 'histogram for GiftAvgAll';  
54     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)  
55     / POSITION = ne;  
56 run;  
57 proc univariate data=WORK.kddcup98 noprint;  
58     histogram GiftAvgCard36;  
59     title 'histogram for GiftAvgCard36';  
60     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)  
61     / POSITION = ne;  
62 run;  
63 proc univariate data=WORK.kddcup98 noprint;  
64     histogram GiftAvgLast;  
65     title 'histogram for GiftAvgLast';  
66     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)  
67     / POSITION = ne;  
68 run;  
69 proc univariate data=WORK.kddcup98 noprint;  
70     histogram GiftCnt36;  
71     title 'histogram for GiftCnt36';  
72     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)  
73     / POSITION = ne;  
74 run;  
75 proc univariate data=WORK.kddcup98 noprint;  
76     histogram GiftCntAll;  
77     title 'histogram for GiftCntAll';  
78     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)  
79     / POSITION = ne;  
80 run;  
81 proc univariate data=WORK.kddcup98 noprint;  
82     histogram GiftCntCard36;  
83     title 'histogram for GiftCntCard36';  
84     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)  
85     / POSITION = ne;  
86 run;  
87 proc univariate data=WORK.kddcup98 noprint;  
88     histogram GiftCntCardAll;  
89     title 'histogram for GiftCntCardAll';  
90     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)  
91     / POSITION = ne;  
92 run;  
93 proc univariate data=WORK.kddcup98 noprint;  
94     histogram GiftTimeFirst;  
95     title 'histogram for GiftTimeFirst';  
96     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)  
97     / POSITION = ne;  
98 run;  
99 proc univariate data=WORK.kddcup98 noprint;  
100     histogram GiftTimeLast;  
101     title 'histogram for GiftTimeLast';  
102     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)  
103     / POSITION = ne;  
104 run;
```

```

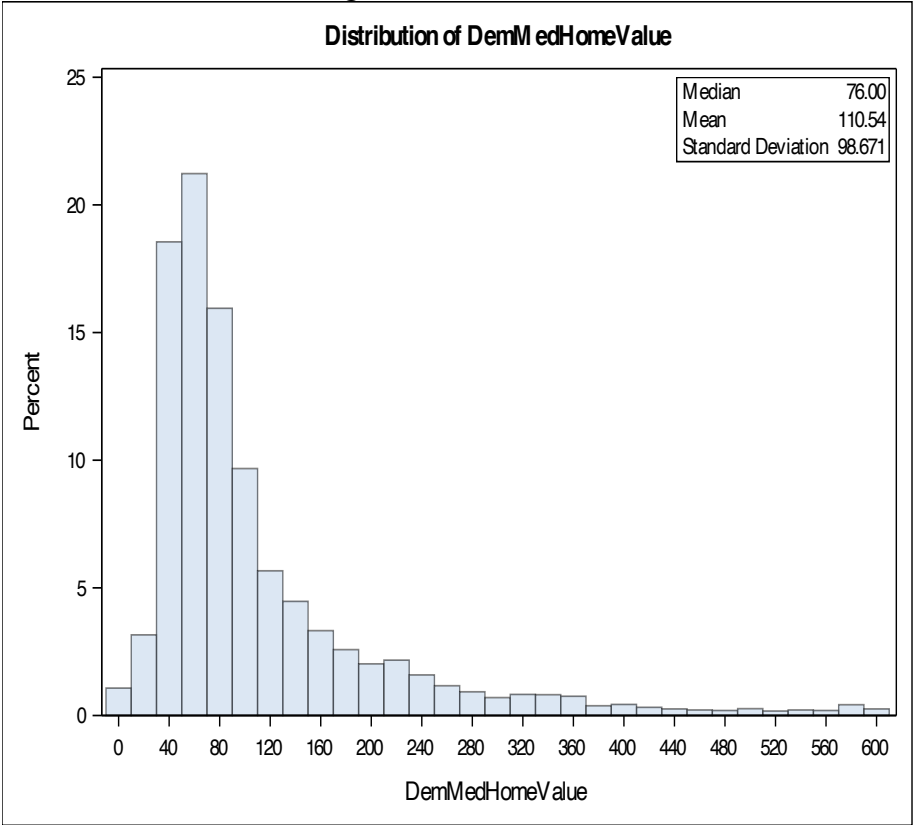
105 proc univariate data=WORK.kddcup98 noprint;
106     histogram PromCnt12;
107     title 'histogram for PromCnt12';
108     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)
109     / POSITION = ne;
110 run;
111 proc univariate data=WORK.kddcup98 noprint;
112     histogram PromCnt36;
113     title 'histogram for PromCnt36';
114     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)
115     / POSITION = ne;
116 run;
117 proc univariate data=WORK.kddcup98 noprint;
118     histogram PromCntAll;
119     title 'histogram for PromCntAll';
120     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)
121     / POSITION = ne;
122 run;
123 proc univariate data=WORK.kddcup98 noprint;
124     histogram PromCntCard12;
125     title 'histogram for PromCntCard12';
126     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)
127     / POSITION = ne;
128 run;
129 proc univariate data=WORK.kddcup98 noprint;
130     histogram PromCntCard36;
131     title 'histogram for PromCntCard36';
132     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)
133     / POSITION = ne;
134 run;
135 proc univariate data=WORK.kddcup98 noprint;
136     histogram PromCntCardAll;
137     title 'histogram for PromCntCardAll';
138     INSET MEDIAN (8.2) MEAN (8.2) STD = 'Standard Deviation' (8.3)
139     / POSITION = ne;
140 run;

```

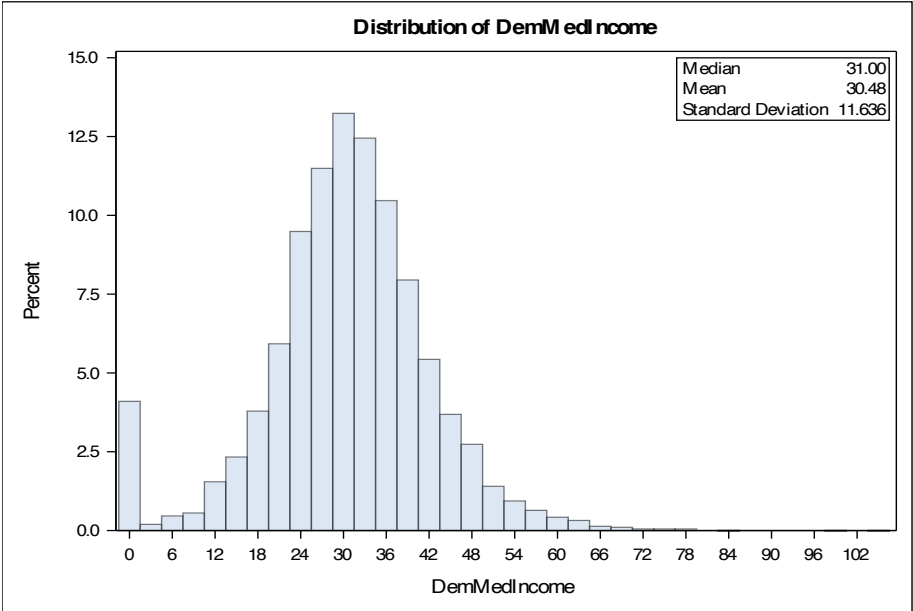
DemAge histogram



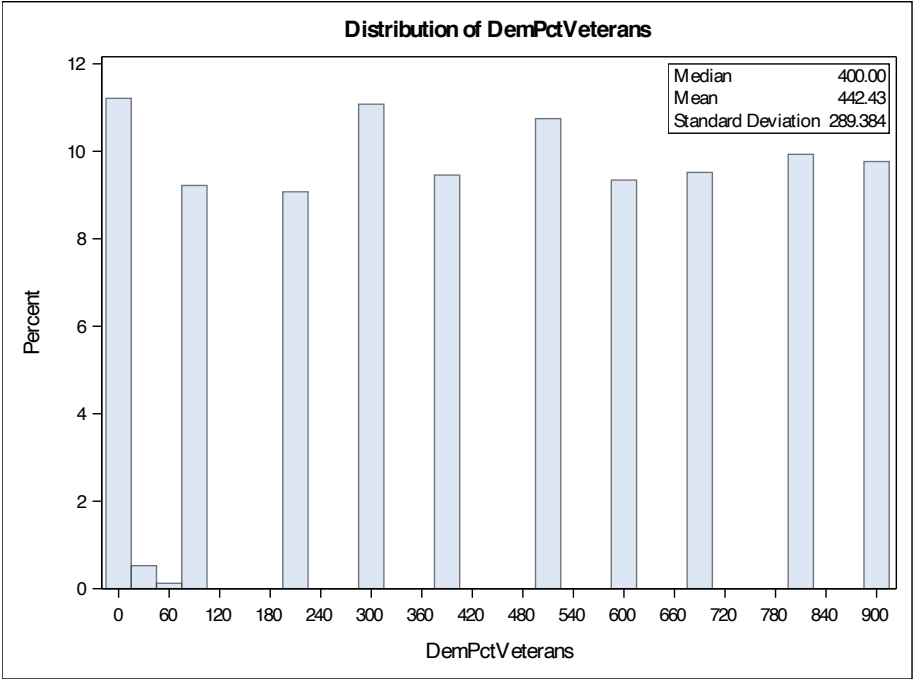
DemMedHomeValue histogram



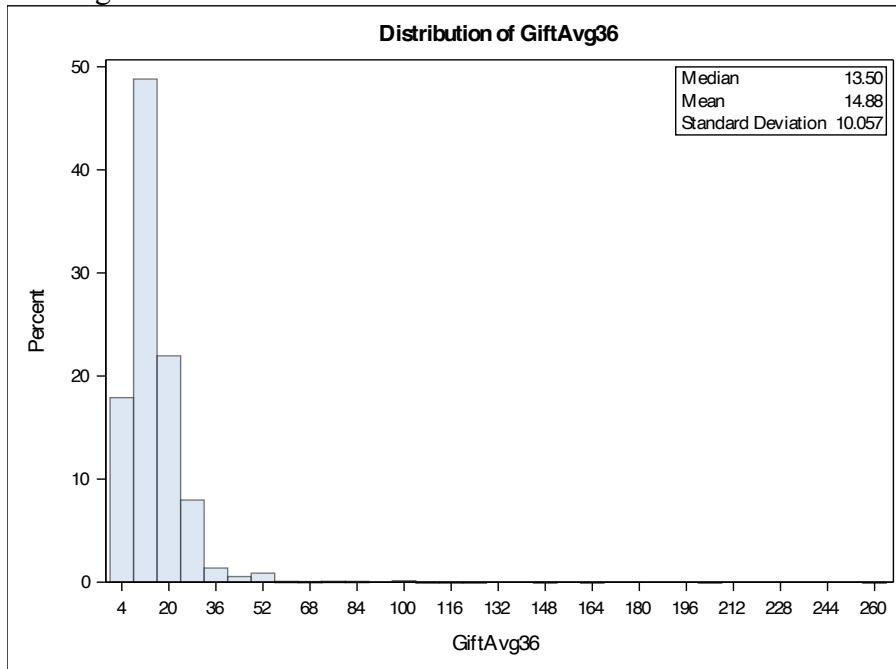
DemMedIncome



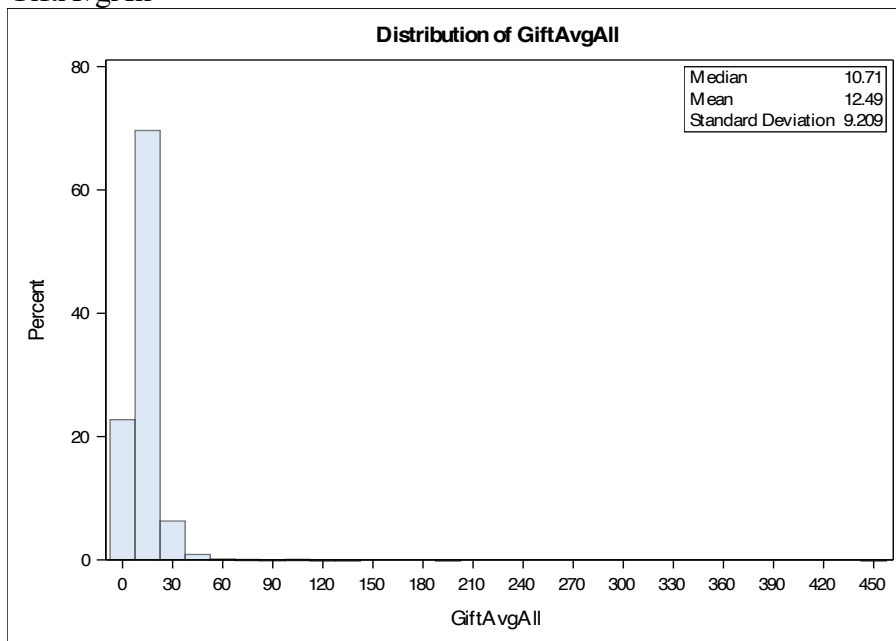
DemPctVeterans



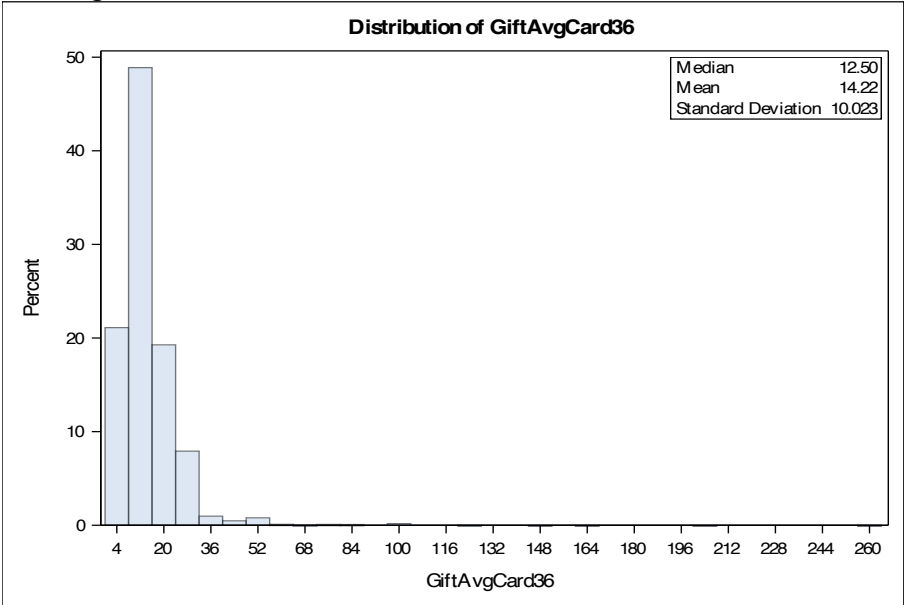
GiftAvg36



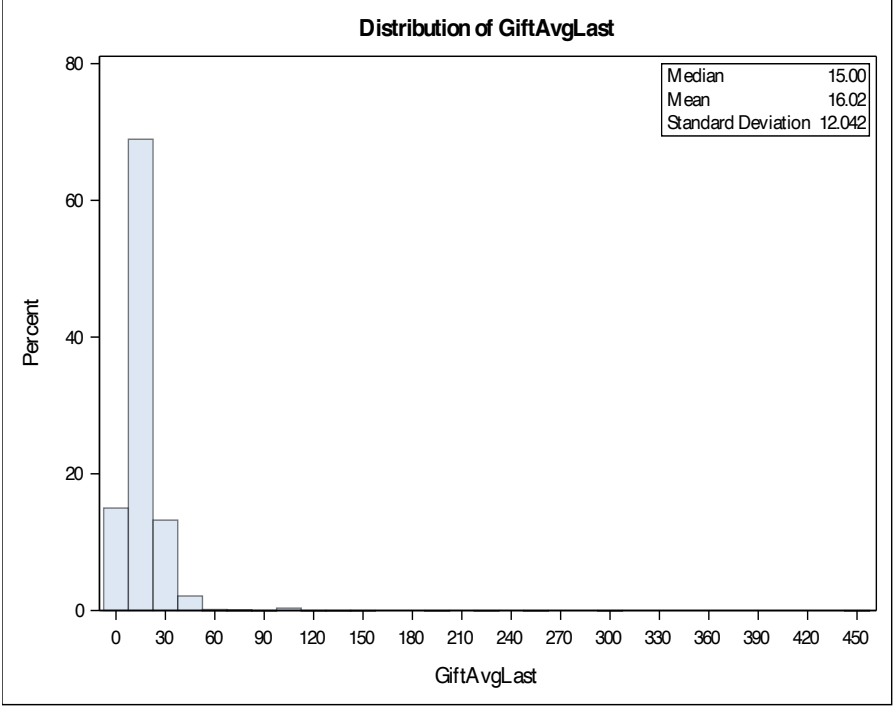
GiftAvgAll



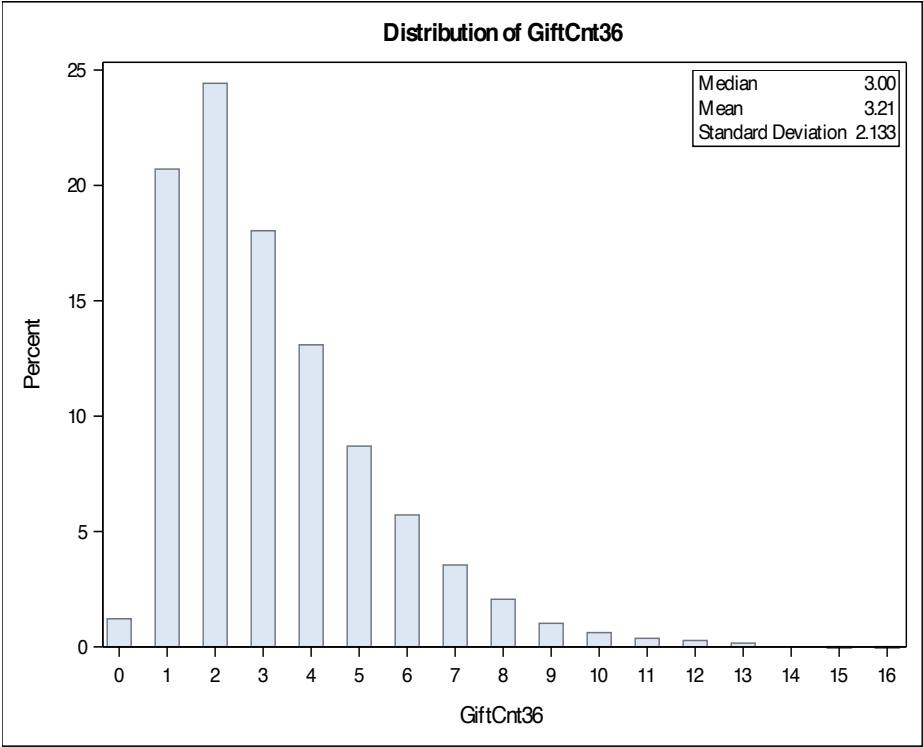
GiftAvgCard36



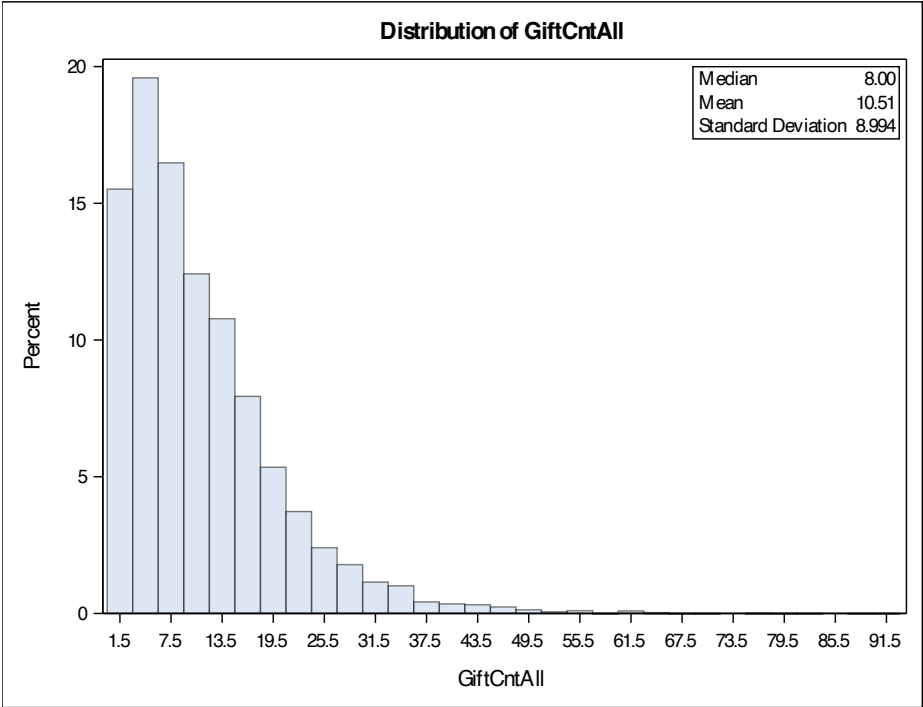
GiftAvgLast



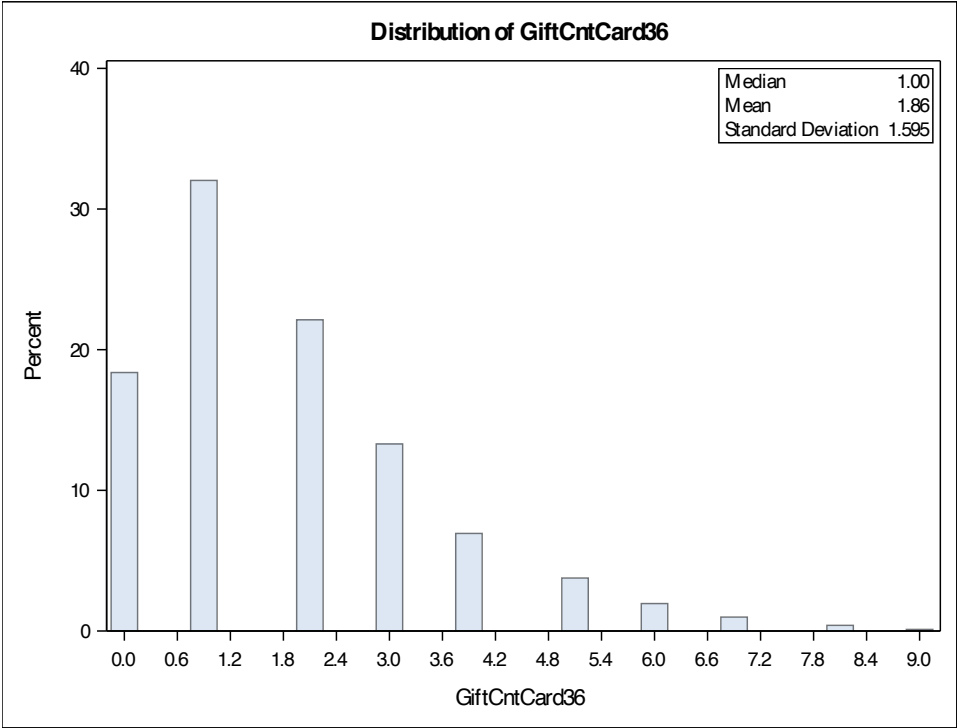
GiftCnt36



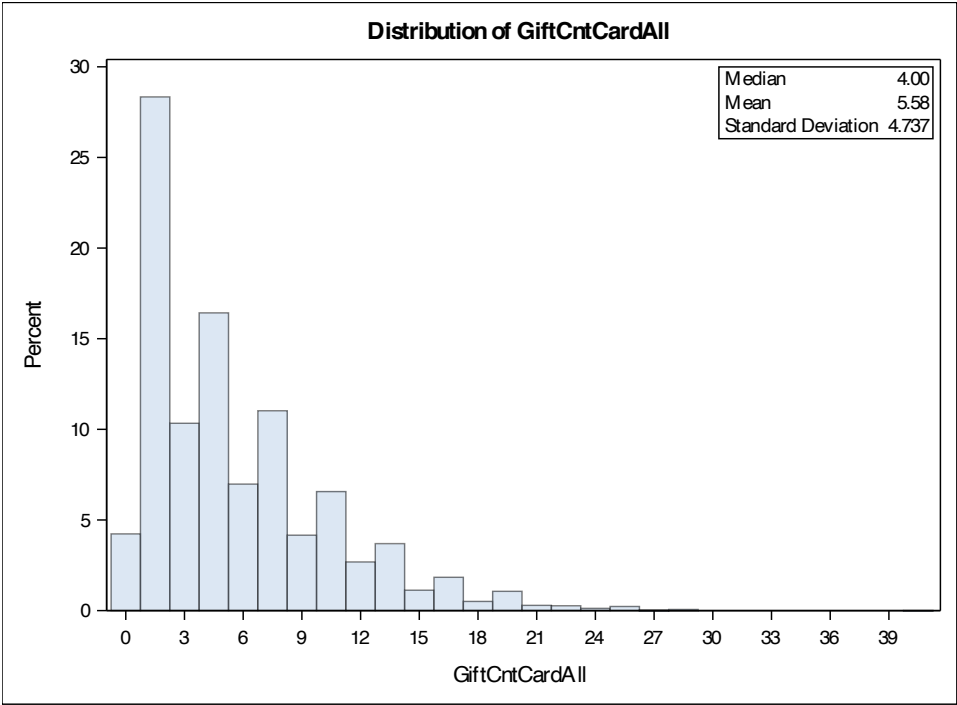
GiftCntAll



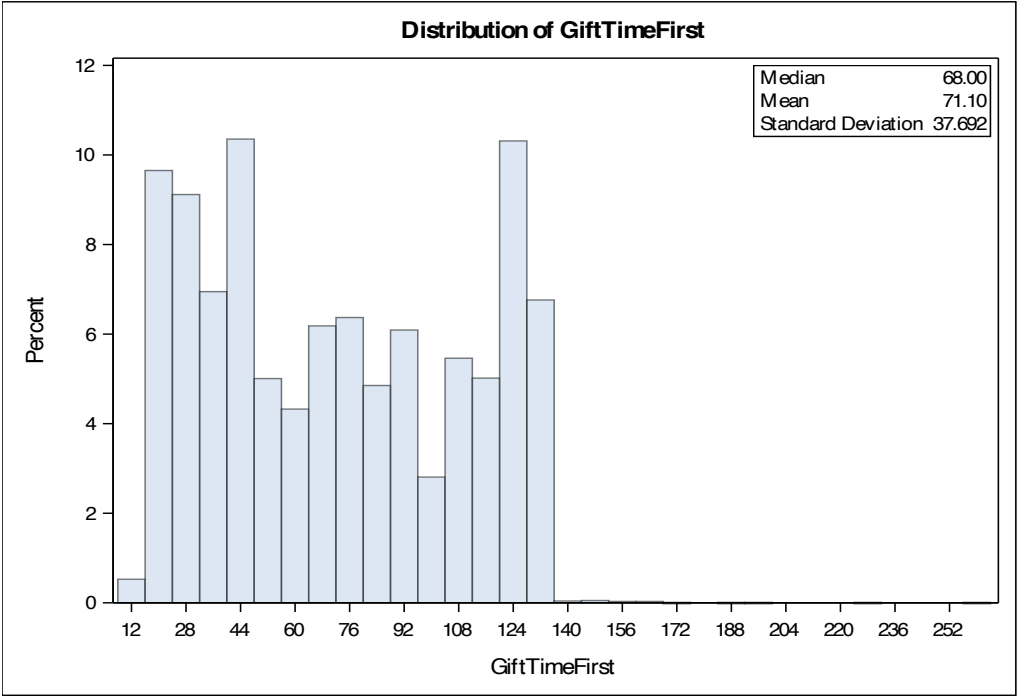
GiftCntCard36



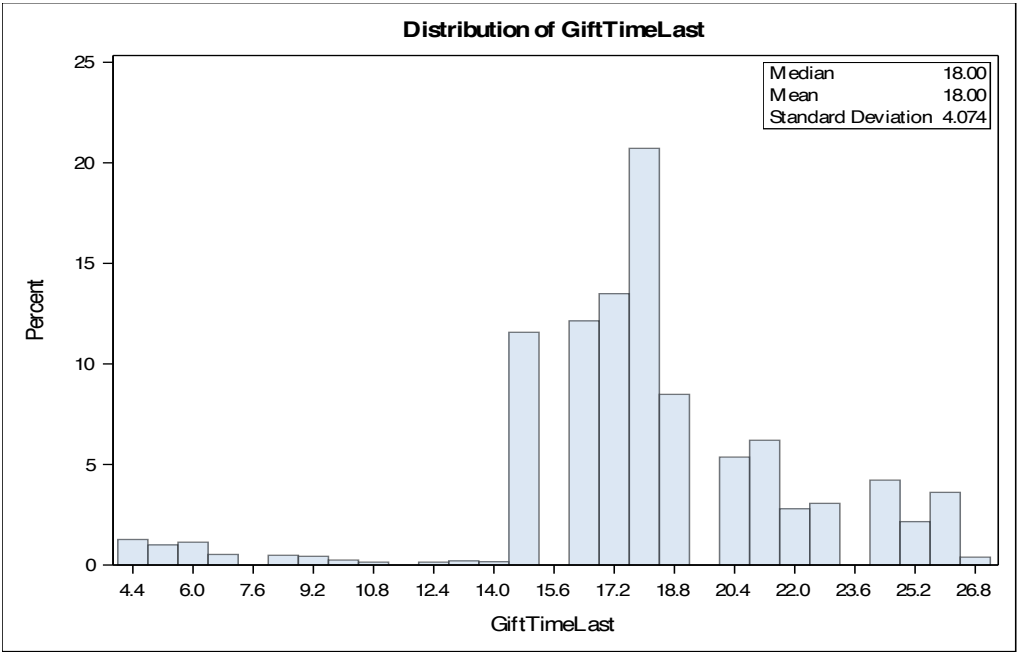
GiftCntCardAll



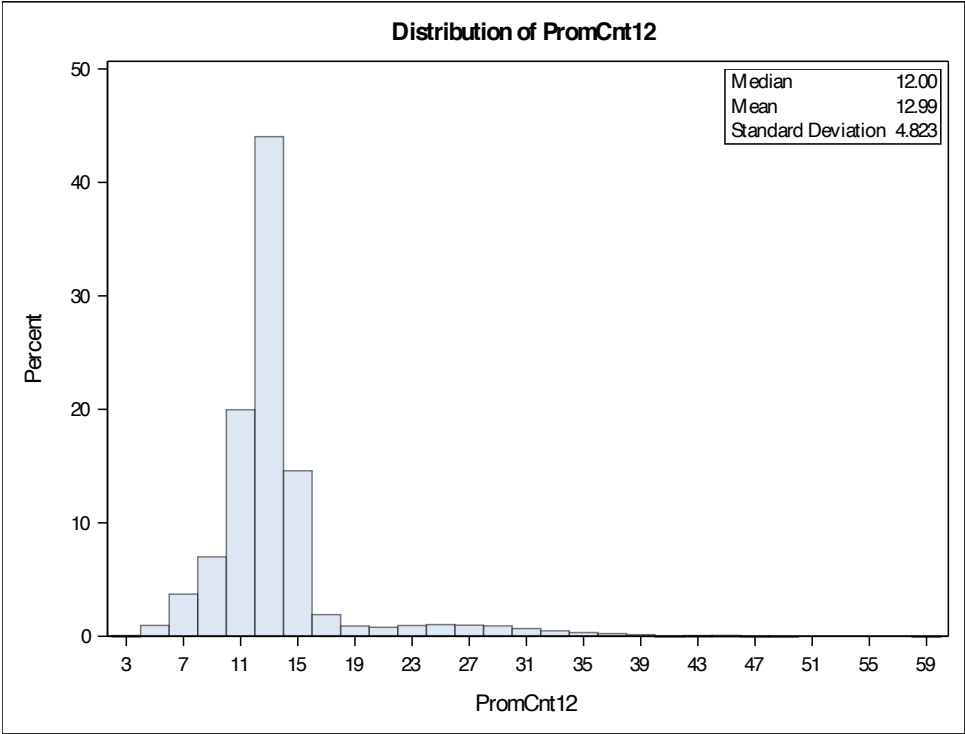
GiftTimeFirst



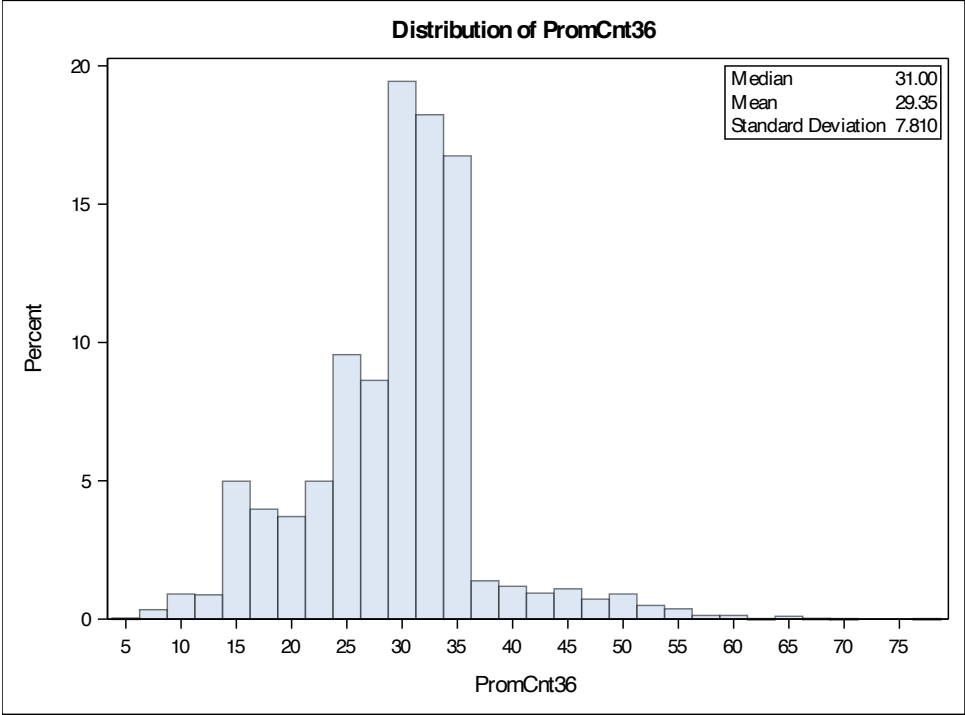
GiftTimeLast



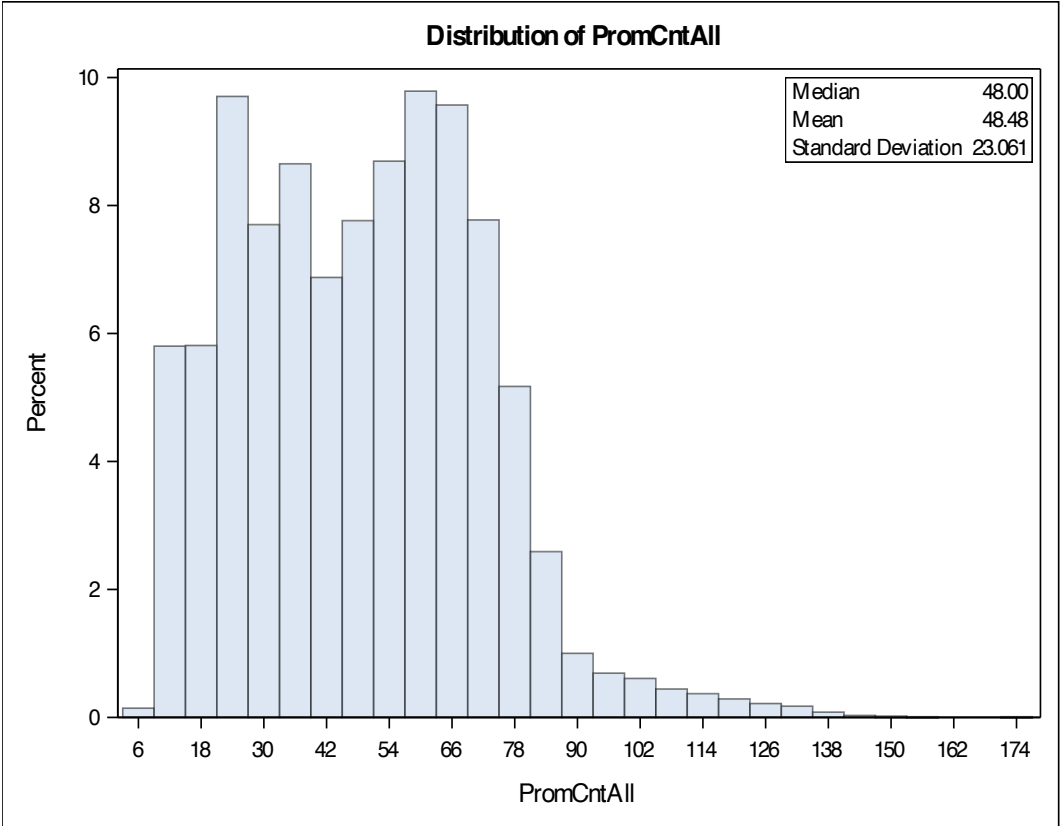
PromCnt12



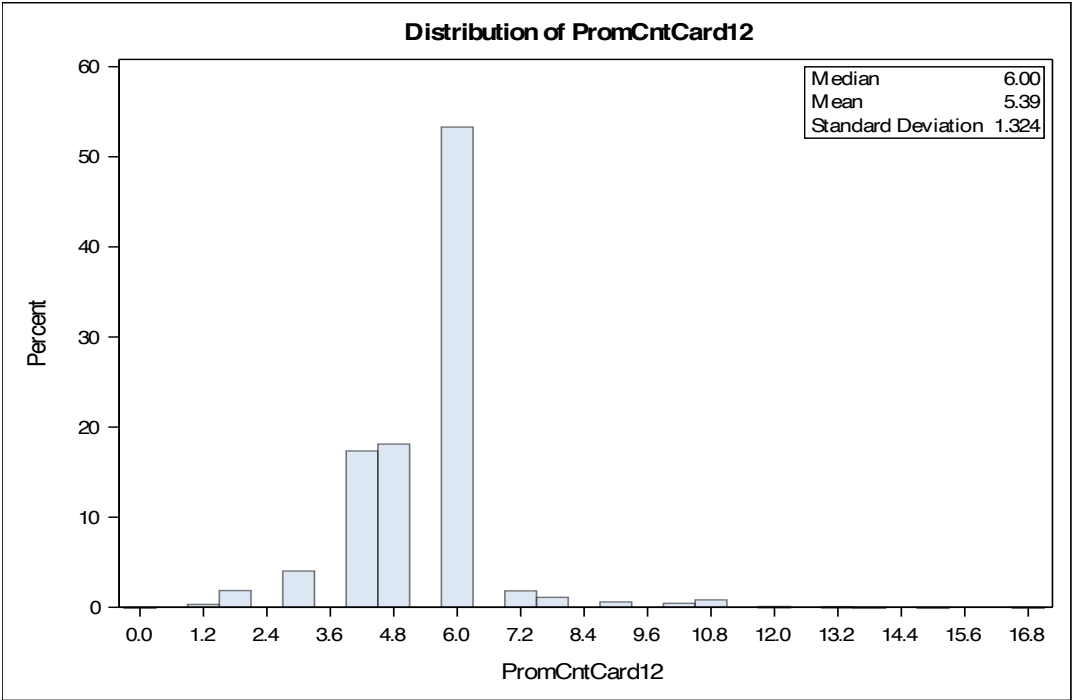
PromCnt36



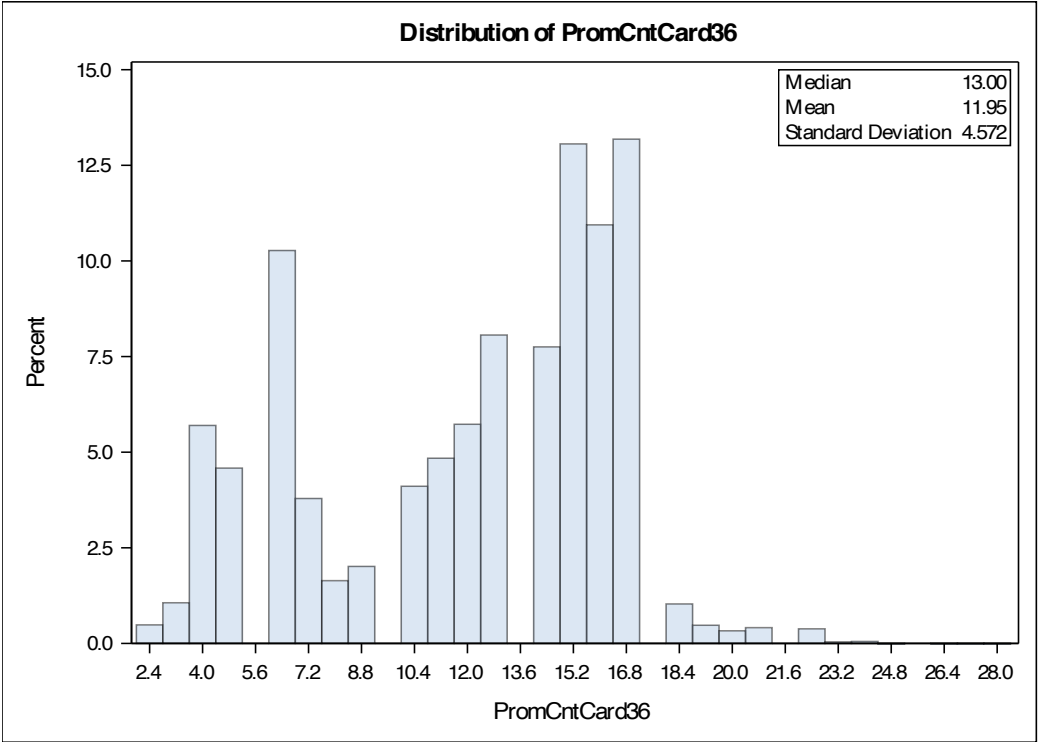
PromCntAll



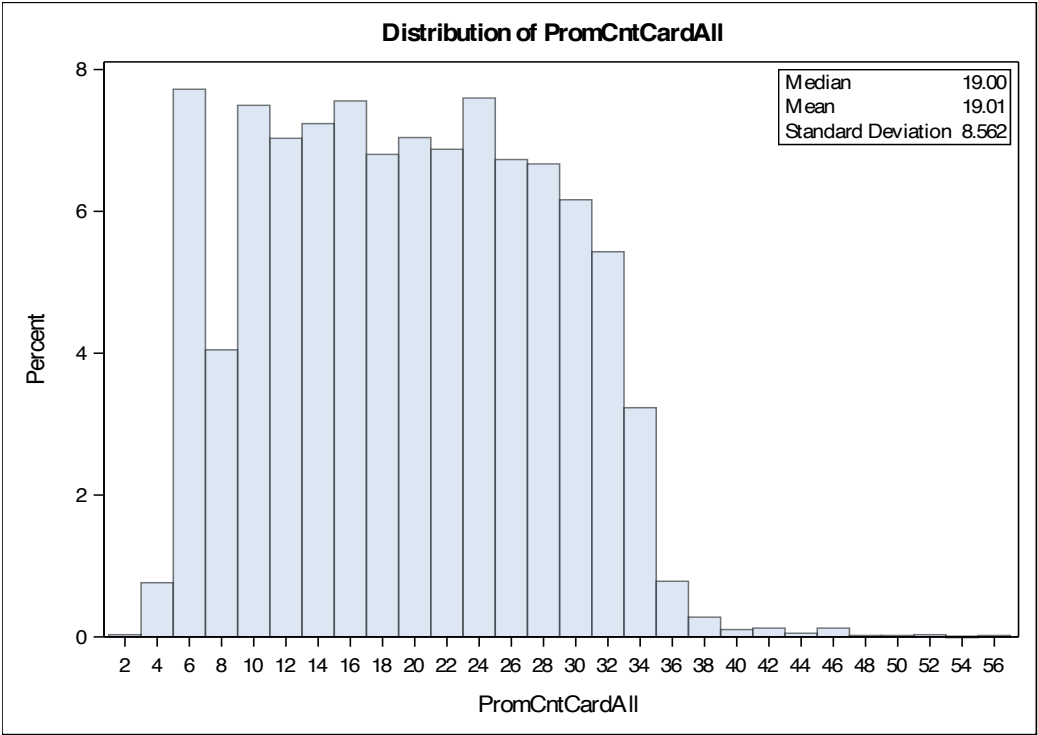
PromCntCard12



PromCntCad36



PromCntCardAll



2.2: Quality check

We first checked the histogram distribution(mean & median) of each numeric variable for anomalies that make no sense, then we proceeded by using the 99 percentile and 1 percentile as our extreme bounds, any value above or below those percentiles will be considered extreme value except for variables with good normal distribution. we found that a lot of the numeric independent variables contain extreme values which caused the distribution of the data to have skewness.

List of variables with unreasonable values

- PromCntCardAll (36,5)
- PromCntCard36 (20,3)
- PromCntCard12 (11,2)
- PromCntAll (114,12)
- PromCnt36 (53,10)
- PromCnt12 (34,5)
- GiftTimeFirst (130,16)
- GiftCntCard36 (7,0)
- GiftCntAll (42,1)
- GiftCnt36 (10,0)
- GiftAvgLast (50,3)
- GiftAvgCard3 (50,3.4)
- GiftAvgAll (40.40,3.33)
- DemMedIncome (60,0)
- DemMedHomeValue (540,0)
- DemAge (87,17)

Code: identify and replacing unreasonable values with missing

```
144 /*2.2 Quality check and replacing extreme values with*/
145
146 ODS select MissingValues;
147 ODS SELECT EXTREMEVALUES;
148 ODS select Quantiles;
149 PROC UNIVARIATE Data= WORK.kddcup98 NEXTRVAL=10;
150 VAR PromCntCardAll
151     PromCntCard36
152     PromCntCard12
153     PromCntAll
154     PromCnt36
155     PromCnt12
156     GiftTimeLast
157     GiftTimeFirst
158     GiftCntCardAll
159     GiftCntCard36
160     GiftCntAll
161     GiftCnt36
162     GiftAvgLast
163     GiftAvgCard36
164     GiftAvgAll
165     DemPctVeterans
166     DemMedIncome
167     DemMedHomeValue
168     DemAge;
169 RUN;
```

```

191 data WORK.kddcup99;
192 set WORK.kddcup98;
193 if PromCntCardAll > 36 then PromCntCardAll = " ";
194 if PromCntCardAll < 5 then PromCntCardAll = " ";
195 if PromCntCard36 > 20 then PromCntCard36 = " ";
196 if PromCntCard36 < 3 then PromCntCard36 = " ";
197 if PromCntCard12 > 11 then PromCntCard12 = " ";
198 if PromCntCard12 < 2 then PromCntCard12 = " ";
199 if PromCntAll > 114 then PromCntAll = " ";
200 if PromCntAll < 12 then PromCntAll = " ";
201 if PromCnt36 > 53 then PromCnt36 = " ";
202 if PromCnt36 < 10 then PromCnt36 = " ";
203 if PromCnt12 > 34 then PromCnt12 = " ";
204 if PromCnt12 < 5 then PromCnt12 = " ";
205 if GiftTimeFirst > 130 then GiftTimeFirst = " ";
206 if GiftTimeFirst < 16 then GiftTimeFirst = " ";
207 if GiftCntCard36 > 7 then GiftCntCard36 = " ";
208 if GiftCntCard36 < 0 then GiftCntCard36 = " ";
209 if GiftCntAll > 42 then GiftCntAll = " ";
210 if GiftCntAll < 1 then GiftCntAll = " ";
211 if GiftCnt36 > 10 then GiftCnt36 = " ";
212 if GiftCnt36 < 0 then GiftCnt36 = " ";
213 if GiftAvgLast > 50 then GiftAvgLast = " ";
214 if GiftAvgLast < 3 then GiftAvgLast = " ";
215 if GiftAvgCard36 > 50 then GiftAvgCard36 = " ";
216 if GiftAvgCard36 < 3.4 then GiftAvgCard36 = " ";
217 if GiftAvgAll > 40.40 then GiftAvgAll = " ";
218 if GiftAvgAll < 3.33 then GiftAvgAll = " ";
219 if DemMedIncome > 60 then DemMedIncome = " ";
220 if DemMedIncome < 0 then DemMedIncome = " ";
221 if DemMedHomeValue > 540 then DemMedHomeValue = " ";
222 if DemMedHomeValue < 0 then DemMedHomeValue = " ";
223 if DemAge > 87 then DemAge = " ";
224 if DemAge < 17 then DemAge = " ";
225 run;
226
--

```

2.3- Numeric independent variables with missing values

- GiftCnt36
- GiftCntAll
- GiftCntCard36
- GiftAvgLast
- GiftAvgAll
- GiftAvgCard36
- GiftTimeFirst
- PromCnt12
- PromCnt36
- PromCntAll
- PromCntCard12
- PromCntCard36
- PromCntCardAll
- DemAge
- DemMedHomeValue
- DemMedIncome

2.4- Right-skewed numeric independent variables

- DemMedHomeValue
- GiftAvg36
- GiftAvgAll
- GiftAvgCard36
- GiftAvgLast
- GiftCnt36
- GiftCntAll
- GiftCntCard36
- GiftCntCardAll
- GiftTimeFirst
- PromCnt12
- PromCntAll
- PromCntCardAll

2.5- Frequency table for the categorical independent variables

Code

```
269 /*2.5 frequency of the categorical data*/
270 proc contents data=WORK.kddcup99;
271 run;
272 proc freq data=WORK.kddcup99;
273     tables DemCluster DemGender DemHomeOwner StatusCat96NK StatusCatStarAll;
274 run;
275
```


DemCluster	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	240	2.48	240	2.48
1	121	1.25	361	3.73
2	191	1.97	552	5.70
3	153	1.58	705	7.28
4	51	0.53	756	7.81
5	95	0.98	851	8.79
6	53	0.55	904	9.33
7	78	0.81	982	10.14
8	182	1.88	1164	12.02
9	70	0.72	1234	12.74
10	175	1.81	1409	14.55
11	236	2.44	1645	16.98
12	323	3.33	1968	20.32
13	309	3.19	2277	23.51
14	248	2.56	2525	26.07
15	108	1.12	2633	27.18
16	201	2.08	2834	29.26
17	178	1.84	3012	31.10
18	321	3.31	3333	34.41
19	50	0.52	3383	34.93
20	171	1.77	3554	36.69
21	165	1.70	3719	38.40
22	125	1.29	3844	39.69
23	131	1.35	3975	41.04
24	401	4.14	4376	45.18
25	135	1.39	4511	46.57
26	100	1.03	4611	47.60
27	331	3.42	4942	51.02
28	194	2.00	5136	53.02
29	73	0.75	5209	53.78
30	262	2.70	5471	56.48
31	125	1.29	5596	57.77
32	72	0.74	5668	58.52
33	52	0.54	5720	59.05
34	132	1.36	5852	60.42
35	384	3.96	6236	64.38
36	401	4.14	6637	68.52
37	99	1.02	6736	69.54
38	118	1.22	6854	70.76
39	242	2.50	7096	73.26
40	432	4.46	7528	77.72
41	197	2.03	7725	79.75
42	140	1.45	7865	81.20

DemCluster	Frequency	Percent	Cumulative Frequency	Cumulative Percent
43	227	2.34	8092	83.54
44	185	1.91	8277	85.45
45	228	2.35	8505	87.81
46	196	2.02	8701	89.83
47	86	0.89	8787	90.72
48	96	0.99	8883	91.71
49	323	3.33	9206	95.04
50	70	0.72	9276	95.77
51	220	2.27	9496	98.04
52	32	0.33	9528	98.37
53	158	1.63	9686	100.00

DemGender	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	5223	53.92	5223	53.92
M	3925	40.52	9148	94.45
U	538	5.55	9686	100.00

DemHomeOwner	Frequency	Percent	Cumulative Frequency	Cumulative Percent
H	5377	55.51	5377	55.51
U	4309	44.49	9686	100.00

StatusCat96N	Frequency	Percent	Cumulative Frequency	Cumulative Percent
K				
A	5826	63.94	5826	63.94
E	227	2.49	6053	66.43
F	660	7.24	6713	73.67
L	34	0.37	6747	74.05
S	2365	25.95	9112	100.00
Frequency Missing = 574				

StatusCatStarAll	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	4450	45.94	4450	45.94
1	5236	54.06	9686	100.00

2.6- Categorical independent variables with missing values

- StatusCat96NK

Step 3: Variable Transformation

3.1- missing value imputation for the categorical independent variables

```
283 /*3.1 missing value imputation for the categorical data*/
284 data WORK.kddcup100;
285     set WORK.kddcup99;
286     if StatusCat96NK=' ' then StatusCat96NK_new="missing";
287     else StatusCat96NK_new=StatusCat96NK;
288     drop StatusCat96NK;
289     rename StatusCat96NK_new=StatusCat96NK;
290 run;
291
292 proc freq data=WORK.kddcup100;
293     tables StatusCat96NK;
294 run;
```

3.2- missing value imputation for the numeric independent variables with missing value indicator

```
296 /*3.2 missing value imputation for numeric independent variables*/
297 /*
298 /*Introducing new missing value indicator variables */
299 data WORK.kddcup101;
300     set WORK.kddcup100;
301     if GiftCnt36 = " " then GiftCnt36_missing =1;
302     else GiftCnt36_missing = 0;
303     if GiftCntAll = " " then GiftCntAll_missing =1;
304     else GiftCntAll_missing = 0;
305     if GiftCntCard36 = " " then GiftCntCard36_missing =1;
306     else GiftCntCard36_missing = 0;
307     if GiftAvgLast = " " then GiftAvgLast_missing =1;
308     else GiftAvgLast_missing = 0;
309     if GiftAvgAll = " " then GiftAvgAll_missing =1;
310     else GiftAvgAll_missing = 0;
311     if GiftAvgCard36 = " " then GiftAvgCard36_missing =1;
312     else GiftAvgCard36_missing = 0;
313     if GiftTimeFirst = " " then GiftTimeFirst_missing =1;
314     else GiftTimeFirst_missing = 0;
315     if PromCnt12 = " " then PromCnt12_missing =1;
316     else PromCnt12_missing = 0;
317     if PromCnt36 = " " then PromCnt36_missing =1;
318     else PromCnt36_missing = 0;
319     if PromCntAll = " " then PromCntAll_missing =1;
320     else PromCntAll_missing = 0;
321     if PromCntCard12 = " " then PromCntCard12_missing =1;
322     else PromCntCard12_missing = 0;
323     if PromCntCard36 = " " then PromCntCard36_missing =1;
324     else PromCntCard36_missing = 0;
325     if PromCntCardAll = " " then PromCntCardAll_missing =1;
326     else PromCntCardAll_missing = 0;
327     if DemAge = " " then DemAge_missing =1;
328     else DemAge_missing = 0;
329     if DemMedHomeValue = " " then DemMedHomeValue_missing =1;
330     else DemMedHomeValue_missing = 0;
331     if DemMedIncome = " " then DemMedIncome_missing =1;
332     else DemMedIncome_missing = 0;
333 run;
```

```

335  /* Imputating missing values of the original independent variables with the mean */
336  proc stdize data=WORK.kddcup101 out=WORK.kddcup101
337      reponly
338      method=MEAN;
339      var
340          GiftCnt36
341          GiftCntAll
342          GiftCntCard36
343          GiftAvgLast
344          GiftAvgAll
345          GiftAvgCard36
346          GiftTimeFirst
347          PromCnt12
348          PromCnt36
349          PromCntAll
350          PromCntCard12
351          PromCntCard36
352          PromCntCardAll
353          DemAge
354          DemMedHomeValue
355          DemMedIncome;
356  run;

```

3.3- Log transformation for the continuous independent variables with right skewed distributions

```

357  /* 3.3 Log transformation*/
358  data WORK.kddcup101;
359      set WORK.kddcup101;
360      if DemMedHomeValue = 0 then DemMedHomeValue = log(0+1);
361      else DemMedHomeValue = log(DemMedHomeValue);
362      if DemPctVeterans = 0 then DemPctVeterans = log(0+1);
363      else DemPctVeterans = log(DemPctVeterans);
364      if GiftAvg36 = 0 then GiftAvg36 = log(0+1);
365      else GiftAvg36 = log(GiftAvg36);
366      if GiftAvgAll = 0 then GiftAvgAll = log(0+1);
367      else GiftAvgAll = log(GiftAvgAll);
368      if GiftAvgCard36 = 0 then GiftAvgCard36 = log(0+1);
369      else GiftAvgCard36 = log(GiftAvgCard36);
370      if GiftCnt36 = 0 then GiftCnt36 = log(0+1);
371      else GiftCnt36 = log(GiftCnt36);
372      if GiftAvgLast = 0 then GiftAvgLast = log(0+1);
373      else GiftAvgLast = log(GiftAvgLast);
374      if GiftCntAll = 0 then GiftCntAll = log(0+1);
375      else GiftCntAll = log(GiftCntAll);
376      if GiftCntCard36 = 0 then GiftCntCard36 = log(0+1);
377      else GiftCntCard36 = log(GiftCntCard36);
378      if GiftCntCardAll = 0 then GiftCntCardAll = log(0+1);
379      else GiftCntCardAll = log(GiftCntCardAll);
380      if GiftTimeFirst = 0 then GiftTimeFirst = log(0+1);
381      else GiftTimeFirst = log(GiftTimeFirst);
382      if PromCnt12 = 0 then PromCnt12 = log(0+1);
383      else PromCnt12 = log(PromCnt12);
384      if PromCntAll = 0 then PromCntAll = log(0+1);
385      else PromCntAll = log(PromCntAll);
386      if PromCntCardAll = 0 then PromCntCardAll = log(0+1);
387      else PromCntCardAll = log(PromCntCardAll);
388  run;
389

```

Step 4: Data partitioning

```
390 /** Step 4 Data partitioning **/  
391 DATA WORK.train WORK.validation;  
392     SET WORK.kddcup101;  
393     RND = RANUNI(20041206);  
394     IF (RND <= .75) then output WORK.train;  
395     else output WORK.validation;  
396 RUN;  
397  
398 /*DATA WORK.train;  
399     SET WORK.kddcup95;  
400     training = RANUNI(75787876);  
401     IF (training <=.75);  
402 RUN;  
403  
404 DATA WORK.validation;  
405     SET WORK.kddcup95;  
406     validation = RANUNI(75787876);  
407     IF (validation <=.25);  
408 RUN;  
409  
410  
411 proc print data=WORK.train(obs=25);  
412 run;  
413 proc contents data=WORK.train;  
414 run;*/  
415
```

Step 5: Stepwise logistic regression for variable selection

```
416 /**Step 5 Step logistic regression**/  
417 PROC LOGISTIC DATA = Work.train;  
418 class DemCluster DemGender DemHomeOwner StatusCat96NK StatusCatStarAll;  
419 MODEL TARGET_B = DemCluster DemGender DemHomeOwner StatusCat96NK StatusCatStarAll  
420 GiftCnt36 GiftCntAll GiftCntCard36 GiftAvgLast GiftAvgAll GiftAvgCard36  
421 GiftTimeFirst PromCnt12 PromCnt36 PromCntAll PromCntCard12 PromCntCard36 PromCntCardAll  
422 DemAge DemMedHomeValue DemMedIncome GiftCnt36_missing GiftCntAll_missing GiftCntCard36_missing  
423 GiftAvgLast_missing GiftAvgAll_missing GiftAvgCard36_missing GiftTimeFirst_missing PromCnt12_missing  
424 PromCnt36_missing PromCntAll_missing PromCntCard12_missing PromCntCard36_missing PromCntCardAll_missing  
425 DemAge_missing DemMedHomeValue_missing DemMedIncome_missing  
426 / selection=stepwise  
427 slentry=0.3  
428 slstay=0.35;  
429 RUN;  
430  
431 /* selected variables:  
432 -GiftCnt36  
433 -GiftAvgLast  
434 -DemMedHomeValue  
435 -StatusCatStarAll  
436 -PromCntCard36_missing  
437 -GiftAvgCard36_missing  
438 -DemMedHomeValue_missg  
439 -DemAge  
440 -PromCntCard36  
441 -PromCntAll  
442 -PromCntCardAll  
443 -GiftAvgAll_missing  
444 -DemAge_missing  
445 -PromCnt12_missing  
446 -DemCluster  
447 */
```

Selected explanatory variables list:

- GiftCnt36
- GiftAvgLast
- DemMedHomeValue
- StatusCatStarAll
- PromCntCard36_missing
- GiftAvgCard36_missing
- DemMedHomeValue_missg
- DemAge
- PromCntCard36
- PromCntAll
- PromCntCardAll
- GiftAvgAll_missing
- DemAge_missing
- PromCnt12_missing
- DemCluster

Step 6: Exporting SAS training and validation datasets

```
449 |
450 | /**Step 6 exporting training and validation datasets**/
451 |
452 | proc export DATA = Work.train
453 |     outfile='/home/u49129236/Amin_Baabol_Homework/INFS762Project/train.csv'
454 |     dbms=csv replace;
455 | run;
456 |
457 | proc export DATA = WORK.validation
458 |     outfile='/home/u49129236/Amin_Baabol_Homework/INFS762Project/validation.csv'
459 |     dbms=csv replace;
460 | run;
461 |
```

Step 7: Using Weka for logistic regression, neural network and support vector machine;

Logistic regression is used for binary classification, the input variables are generally numeric/nominal. This algorithm learns a coefficient for all the explanatory covariates which are then combined into a regression function. Neural network is a classifier that uses backpropagation to learn a multi-layer perceptron to classify instances. Lastly, SVM is also called maximum margin classifier because it draws a line between positive and negative examples and the maximum margin is found which then prevents overfitting.

TARGET_B=1	Logistic Regression	Neural Network	SVM(SMO)
Precision	0.567	0.556	0.531
Recall	0.599	0.613	0.578
Accuracy	58.5448%	57.665%	55.0338%

Evidently, all three models have similar accuracies, however, for this particular case logistic regression seems to be outperforming the other two algorithms just slightly. We, therefore, recommend logistic regression.

Task 2: Regression (with TARGET_D as the dependent variable)

Task-2

Step 1 code

```
/* Step-1 removing missing values for TARGET_D in from the original data*/  
data WORK.kddcup101;  
set WORK.kddcup101;  
if TARGET_D = ' ' then delete;  
run;
```

Step 2 (w)

Training dataset is the portion of the data that is used to fit the model. This data is the data that is feed to the model to train and learn.

Validation dataset is the portion of the data used to give an unbiased evaluation of a model and then fine tune parameters. In the industry, validation dataset is used to fine tune the model hyperparameters. The validation dataset result's is used to update parameters.

Test dataset is the portion of the data used to evaluate the model after the model was trained using the train and validation datasets. Usually, this dataset is used to evaluate different models.

Validation:

For this method, a data set is divided into three data sets training, validation, and testing. We use the training dataset to train the model then we use the validation dataset to test the model and choose the hyperparameters that performed the best on the validation dataset and finally we test the model using the test dataset.

Cross-Validation:

This method divides the dataset into more than one split. It can divide the dataset into 3,5,10 or any k number of splits. The method builds multiple models and for each model it uses some folds train the model and the rest to test the model.

Step 3 code

```
/* Dropping the variable Demcluster */
```

```
DATA WORK.kddcup101;  
SET WORK.kddcup101;  
drop DemCluster;  
RUN;
```

```
3 /* creating dummy variables */  
4  
5 data WORK.kddcup101;  
6   set WORK.kddcup101;  
7   IF StatusCat96NK = 'missing' THEN StatusCat96NK_Missing = 1;  
8   ELSE StatusCat96NK_Missing = 0;  
9   IF DemGender = 'F' THEN DemGender_F = 1;  
10  ELSE DemGender_F = 0;  
11  IF DemGender = 'M' THEN DemGender_M = 1;  
12  ELSE DemGender_M = 0;  
13  IF DemGender = 'U' THEN DemGender_U = 1;  
14  ELSE DemGender_U = 0;  
15  IF DemHomeOwner = 'H' THEN DemHomeOwner_H = 1;  
16  ELSE DemHomeOwner_H = 0;  
17  IF DemHomeOwner = 'U' THEN DemHomeOwner_U = 1;  
18  ELSE DemHomeOwner_U = 0;  
19  IF StatusCat96NK = 'A' THEN StatusCat96NK_A = 1;  
20  ELSE StatusCat96NK_A = 0;  
21  IF StatusCat96NK = 'E' THEN StatusCat96NK_E = 1;  
22  ELSE StatusCat96NK_E = 0;  
23  IF StatusCat96NK = 'F' THEN StatusCat96NK_F = 1;  
24  ELSE StatusCat96NK_F = 0;  
25  IF StatusCat96NK = 'S' THEN StatusCat96NK_S = 1;  
26  ELSE StatusCat96NK_S = 0;  
27  IF StatusCat96NK = 'L' THEN StatusCat96NK_L = 1;  
28  ELSE StatusCat96NK_L = 0;  
29  
30 run;  
31  
32 /* dropping original categorical independent variables whom we have created dummy variables  
33 for and dropping the L dummy variable to avoid dummy trap */  
34 data WORK.kddcup101;  
35   set WORK.kddcup101;  
36   drop StatusCat96NK_Missing;  
37 run;  
38
```

Step 4

- GiftAvg36
- GiftAvgLast_missing
- GiftAvgLast
- GiftAvgAll_missing
- PromCnt36_missing
- GiftTimeFirst
- GiftAvgCard36_missing
- PromCntCard36
- DemHomeOwner_H
- DemGender_F
- GiftAvgAll
- PromCntAll_missing
- GiftCntCard36_missing
- PromCntCard12
- DemMedHomeValue_missing
- GiftTimeFirst_missing
- DemPctVeterans
- StatusCat96NK_F
- GiftCntAll
- GiftCntCard36

Step 5

1. Linear Regression Model

RMSE = 9.3016

```
Linear Regression Model

TARGET_D =

-0.5641 * GiftCntAll +
0.4492 * GiftCntCard36 +
5.1619 * GiftAvgLast +
8.1496 * GiftAvg36 +
1.0249 * GiftAvgAll +
-0.5077 * GiftTimeFirst +
0.1689 * PromCntCard36 +
13.9904 * GiftAvgLast_missing +
10.7804 * GiftAvgAll_missing +
2.6708 * GiftAvgCard36_missing +
3.5522 * PromCnt36_missing +
1.7323 * PromCntAll_missing +
-0.6181 * DemGender_F +
-0.7601 * DemHomeOwner_H +
-19.2949

Time taken to build model: 0.02 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient          0.6643
Mean absolute error             4.9027
Root mean squared error         9.3016
Relative absolute error         64.1897 %
Root relative squared error     74.741 %
Total Number of Instances      4843
```

2. k nearest neighbor (KNN)

RMSE = 12.5779

```
=== Classifier model (full training set) ===

IB1 instance-based classifier
using 1 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient          0.4707
Mean absolute error             6.3116
Root mean squared error         12.5779
Relative absolute error         82.6375 %
Root relative squared error     101.0675 %
Total Number of Instances      4843
```

3. Support Vector Regression

RMSE = 9.897

RMRreg

```
weights (not support vectors):
- 0.0158 * (normalized) GiftCntAll
+ 0.0001 * (normalized) GiftCntCard36
+ 0.063 * (normalized) GiftAvgLast
+ 0.1244 * (normalized) GiftAvg36
+ 0.0317 * (normalized) GiftAvgAll
+ 0.0033 * (normalized) GiftTimeFirst
+ 0.0011 * (normalized) PromCntCard12
+ 0.0094 * (normalized) PromCntCard36
- 0.0022 * (normalized) DemPotVeterans
- 0.0012 * (normalized) GiftCntCard36_missing
+ 0.0002 * (normalized) GiftAvgLast_missing
+ 0.0019 * (normalized) GiftAvgAll_missing
+ 0.0091 * (normalized) GiftAvgCard36_missing
- 0.0020 * (normalized) GiftTimeFirst_missing
- 0.0017 * (normalized) PromCnt36_missing
+ 0.0062 * (normalized) PromCntAll_missing
+ 0.0017 * (normalized) DemMedHomeValue_missing
- 0.0005 * (normalized) DemGender_F
- 0.0007 * (normalized) DemHomeOwner_R
- 0.0033 * (normalized) StatusCat96HK_F
- 0.0357

Number of kernel evaluations: 536396423 (49.537% cached)

Time taken to build model: 106.62 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient      0.6285
Mean absolute error        4.5467
Root mean squared error    9.897
Relative absolute error    59.529 %
Root relative squared error 79.5257 %
Total Number of Instances  4843
```

which model gives you the best RMSE?)

Linear regression model gives the best RMSE value.