Gavin Gunawardena
INFS 762
Dr. Liu
Project 1

1.SAS Code:

/* Pre Setup for Project */

```
/* Assign library */

libname PROJ1 '/home/u58783680/INFS 762 Data Warehousing/Project 1';

/* Step 1 */


/*Import organics.csv*/

proc import datafile='/home/u58783680/INFS 762 Data Warehousing/Project 1/organics.csv'
out=PROJ1.Organics   dbms=CSV    replace;

/* delimiter='09'x; */

 getnames=yes;

run;


/* Remove variables: DemCluster and TargetAmount (correction: TargetAmt) */

DATA PROJ1.Organics;

SET PROJ1.Organics;

drop DemCluster TargetAmt;

RUN;
```

/* Step 2 */

/* Create histograms of continuous variables in dataset.

```
The continuous variables are: PromSpend, DemAffl, DemAge, PromTime

*/

goptions reset=global

      gunit=pct

      hsize= 10.625 in

      vsize= 8.5 in

      htitle=4
```

```
        htext=3
        vorigin=0 in
        horigin= 0 in
        cback=white border
        ctext=black
        colors=(black blue green red yellow)
        ftext=swiss
        lfactor=3;


proc univariate data=PROJ1.Organics;
  histogram PromSpend DemAffl DemAge PromTime;
  var  PromSpend DemAffl DemAge PromTime;
/*    title 'Histograms for Continuous Variables in Dataset'; */
run;


/* *Modified the proc univariate function to show stats to confirm skewness. PromTime, PromSpend,
and DemAffl have skewed distributions. */

/* Check for extreme and missing values */
ODS SELECT EXTREMEVALUES;
ODS select MissingValues;




PROC UNIVARIATE Data=PROJ1.Organics NEXTRVAL=10;
VAR DemAffl DemAge PromSpend PromTime;
RUN;
```

/* Note: Going by these results DemAffl does not make sense since it has values outside of the 1-30 range, */

/* including 0, 31, and 34 multiple times. */

/* Note: PromSpend of .01 seems to be very common. .01 could be being used as a placeholder for 0 or maybe promotions */

/* with a cost of .01 is common. */

/* Check for the amount of categories in each of the categorical variables and find missing values*/

```
PROC freq Data=PROJ1.Organics;

table DemClusterGroup DemGender DemReg DemTVReg PromClass;

RUN;
```

/* Step 3 */

/* Create dummy code for PromClass */

```
proc iml;

USE PROJ1.Organics;

  read all var "PromClass";

close;


uPromClass = UNIQUE(PromClass);

print uPromClass;

run;



data PROJ1.Organics;

 SET PROJ1.Organics ;


 IF PromClass = 'Platinum' THEN PromClass_Dummy_Platinum = 1;

   ELSE PromClass_Dummy_Platinum = 0;
```

```
 IF PromClass = 'Gold' THEN PromClass_Dummy_Gold = 1;
   ELSE PromClass_Dummy_Gold = 0;
 IF PromClass = 'Silver' THEN PromClass_Dummy_Silver = 1;
   ELSE PromClass_Dummy_Silver = 0;
RUN;

PROC FREQ DATA=PROJ1.Organics;
 TABLES PromClass_Dummy_Gold*PromClass_Dummy_Platinum*PromClass_Dummy_Silver / list ;
RUN;
```

/* Step 4 */

/* Missing Value Imputation for the continuous, discrete and categorical variables */

```
/* Continuous  (replace with mean and add indicator in additional column)*/
/* PromSpend - No missing values*/

/* Impute the missing values of DemAffl, DemAge, and PromTime via finding the means of these features, putting them into macro variables, */
/* and replacing the unknown values with them. Also, create a new column that indicates whether the value was missing. Rounded the means
to whole numbers to match the rest of the data in these three columns.*/
proc iml;
use PROJ1.Organics;
read all;
median_DemAffl = median(DemAffl);
call symput("median_DemAffl",rowcat(char(median_DemAffl)));
 %put &median_DemAffl;
median_DemAge = median(DemAge);
```

```
call symput("median_DemAge",rowcat(char(median_DemAge)));
 %put &median_DemAge;
median_PromTime = median(PromTime);
call symput("median_PromTime",rowcat(char(median_PromTime)));
 %put &median_PromTime;
quit;

DATA PROJ1.Organics;
SET PROJ1.Organics;
/* •	DemAffl (interval, possibly MNAR)*/
IF missing(DemAffl) THEN DO;
	DemAffl_M=1;
	DemAffl="&median_DemAffl";
END;
ELSE DO;
DemAffl_M=0;
END;
/* •	DemAge (interval, possibly MNAR)*/
IF missing(DemAge) THEN DO;
	DemAge_M=1;
	DemAge="&median_DemAge";
END;
ELSE DO;
DemAge_M=0;
END;
/* •	PromTime (nominal, MAR) */
IF missing(PromTime) THEN DO;
	PromTime_M=1;
	PromTime="&median_PromTime";
```

```
END;
ELSE DO;
PromTime_M=0;
END;
RUN;


/* •	DemAge (interval, possibly MNAR) */
PROC LOGISTIC DATA= PROJ1.Organics;
class DemGender DemReg DemTVReg PromClass DemClusterGroup;
model DemAge=  DemAffl PromSpend PromTime  ;
RUN;
/* •	PromTime (nominal, MAR) */
/* Categorical (create unknown category for missing and add indicator in additional column) */
/* •	DemClusterGroup */
DATA PROJ1.Organics;
SET PROJ1.Organics;
IF missing(DemClusterGroup) THEN DO;
	DemClusterGroup_M=1;
	DemClusterGroup="U";
END;
ELSE DO;
DemClusterGroup_M=0;
END;
RUN;
/* •	DemGender */
DATA PROJ1.Organics;
SET PROJ1.Organics;
IF missing(DemGender) THEN DO;
```

```
        DemGender_M=1;
        DemGender="U";
END;
ELSE DO;
DemGender_M=0;
END;
RUN;
/* •      DemReg */
DATA PROJ1.Organics;
SET PROJ1.Organics;
IF missing(DemReg) THEN DO;
        DemReg_M=1;
        DemReg="Unknown";
END;
ELSE DO;
DemReg_M=0;
END;
RUN;
/* •      DemTVReg */
DATA PROJ1.Organics;
SET PROJ1.Organics;
IF missing(DemTVReg) THEN DO;
        DemTVReg_M=1;
        DemTVReg="Unknown";
END;
ELSE DO;
DemTVReg_M=0;
END;
RUN;
```

/* Step 5 */

/* Split the dataset between Training and Validation */

```
Data PROJ1.Organics_Training PROJ1.Organics_Validation;

        set PROJ1.Organics;

        RND = ranuni(20053206);

        if (RND <= .6) then output PROJ1.Organics_Training;

        else output PROJ1.Organics_Validation;

Run;
```

/* Step 6 */

/* Use stepwise logistic regression for variable selection and export the datasets */

```
PROC LOGISTIC DATA= PROJ1.Organics_Training;

class  DemGender DemReg DemTVReg PromClass_Dummy_Platinum PromClass_Dummy_Gold
PromClass_Dummy_Silver DemAffl_M DemAge_M DemReg_M DemTVReg_M PromTime_M
DemClusterGroup_M DemGender_M;

model TargetBuy= DemAffl DemAge PromSpend PromTime DemGender DemReg DemTVReg
PromClass_Dummy_Platinum PromClass_Dummy_Gold PromClass_Dummy_Silver DemAffl_M
DemAge_M DemReg_M DemTVReg_M PromTime_M DemClusterGroup_M DemGender_M /
selection=stepwise;

RUN;

/* The logistic function with stepwise variable selection ended up choosing: */

/* DemAffl, DemAge, DemGender and their respective missing value indicators (DemAffl_M,
DemAge_M, DemGender_M) as optimal variables for predicting TargetBuy */
```

/* Export the datasets */

```
proc export data=PROJ1.Organics

  outfile='/home/u58783680/INFS 762 Data Warehousing/Project 1/organics_s6_export.csv'

  dbms=csv replace;

run;

proc export data=PROJ1.Organics_Training
```

```
  outfile='/home/u58783680/INFS 762 Data Warehousing/Project 1/organics_s6_Training.csv'
  dbms=csv replace;
run;
proc export data=PROJ1.Organics_Validation
  outfile='/home/u58783680/INFS 762 Data Warehousing/Project 1/organics_s6_Validation.csv'
  dbms=csv replace;
run;
```

/* Step 7 */

/* Fit the model into 3 different algorithms in Weka */

```
/* Done in Weka */
```

/* Step 8 */

/* Transform the identified skewed variables via a log transformation */

```
DATA PROJ1.Organics_s8;
SET PROJ1.Organics;
promSpend_Log = log(promSpend+1);
promTime_Log = log(promTime+1);
DemAffl_Log = log(DemAffl+1);
Run;

/* View change in distribution */
goptions reset=global
      gunit=pct
      hsize= 10.625 in
      vsize= 8.5 in
      htitle=4
      htext=3
```

```
        vorigin=0 in
        horigin= 0 in
        cback=white border
        ctext=black
        colors=(black blue green red yellow)
        ftext=swiss
        lfactor=3;


proc univariate data=PROJ1.Organics_s8;
  histogram promSpend_Log DemAffl_Log DemAge promTime_Log;
    var  promSpend_Log DemAffl_Log DemAge promTime_Log;
/*    title 'Histograms for Continuous Variables in Dataset'; */
run;
```

/* Step 9 */

/* Split the dataset between Training and Validation */

```
Data PROJ1.Organics_s9_Training PROJ1.Organics_s9_Validation;
            set PROJ1.Organics_s8;
            RND = ranuni(20053206);
            if (RND <= .6) then output PROJ1.Organics_s9_Training;
            else output PROJ1.Organics_s9_Validation;
Run;


/* Use the stepwise variable selection method with the transformed training dataset */
PROC LOGISTIC DATA= PROJ1.Organics_s9_Training;
```

```
class  DemGender DemReg DemTVReg PromClass_Dummy_Platinum PromClass_Dummy_Gold
PromClass_Dummy_Silver DemAffl_M DemAge_M DemReg_M DemTVReg_M PromTime_M
DemClusterGroup_M DemGender_M;

model TargetBuy= DemAffl_Log DemAge promSpend_Log promTime_Log DemGender DemReg
DemTVReg PromClass_Dummy_Platinum PromClass_Dummy_Gold PromClass_Dummy_Silver
DemAffl_M DemAge_M DemReg_M DemTVReg_M PromTime_M DemClusterGroup_M DemGender_M
/ selection=stepwise;

RUN;

/* The logistic function with stepwise variable selection ended up choosing DemAffl_Log, DemGender,
DemAge, DemAffl_M, DemAge_M, and DemGender_M.*/
```

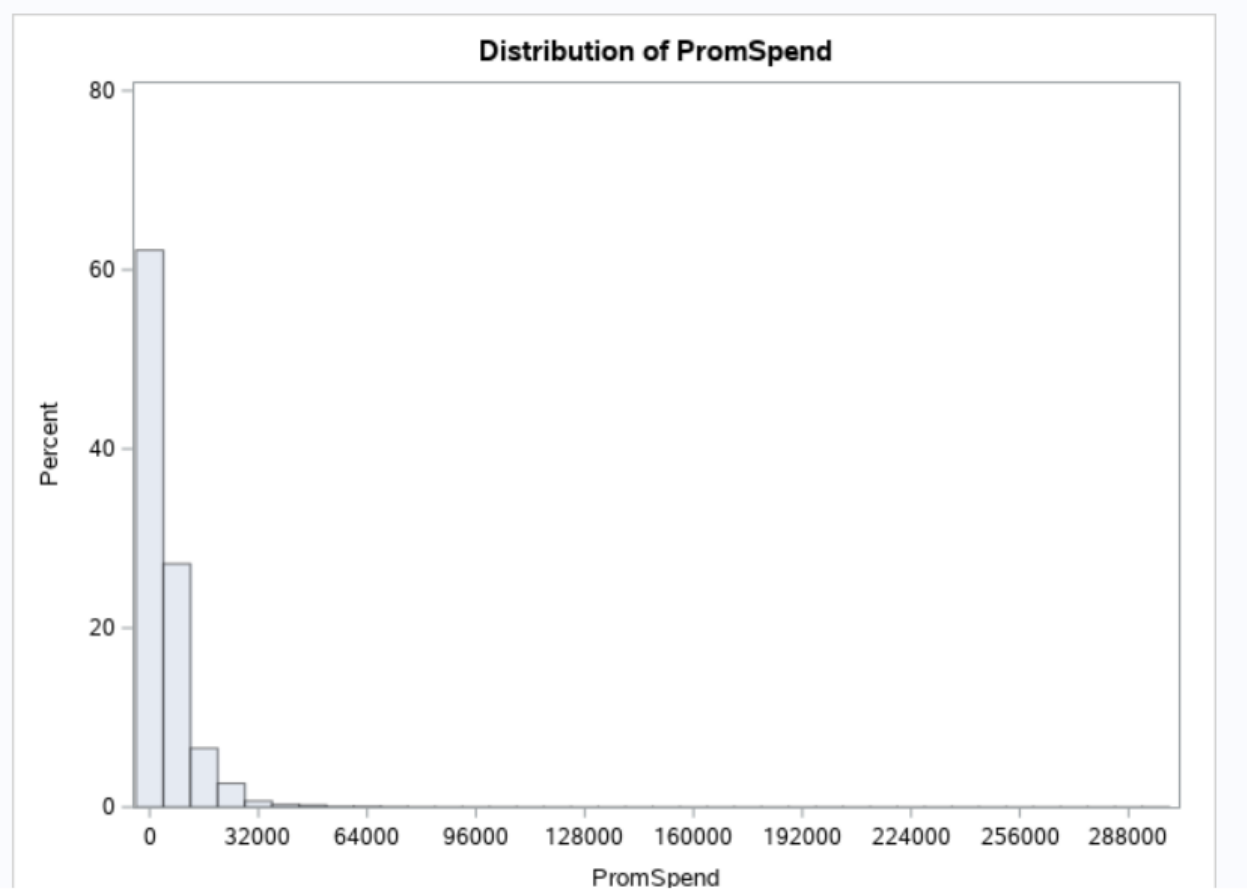
/* Step 10 */

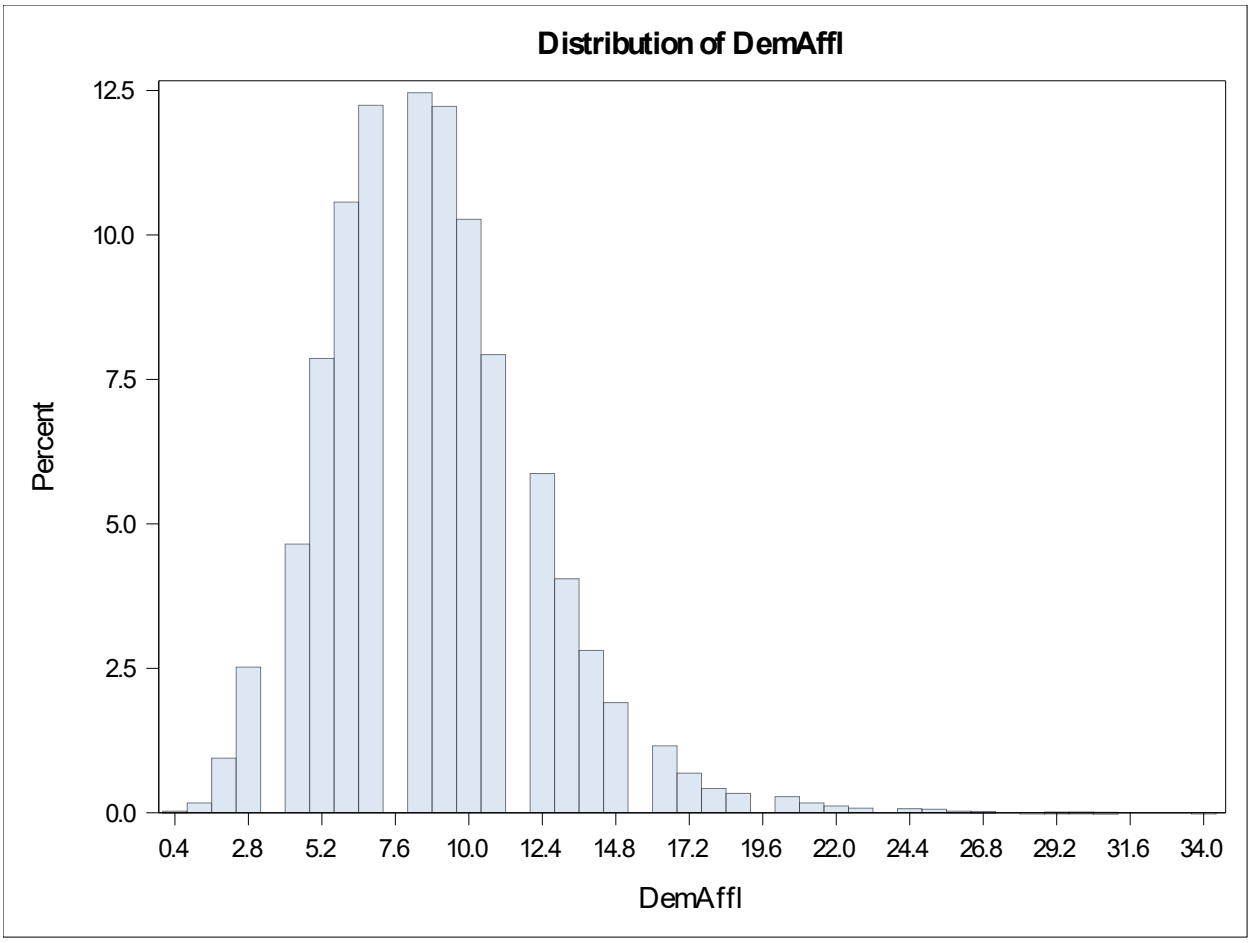
/* Fit the same three models as previously done in step 7 but with the log transformed dataset*/

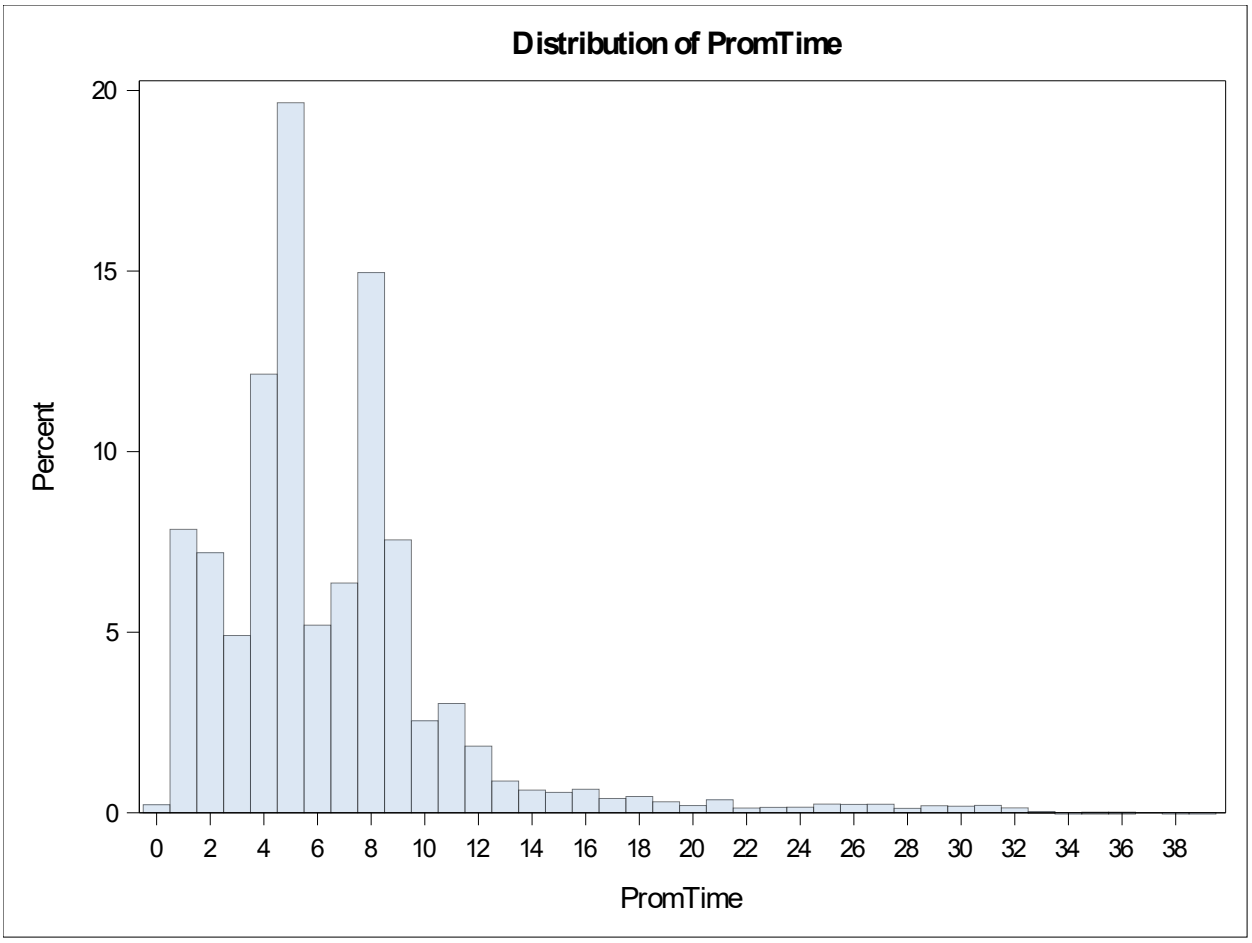
```
/* Export the updated datasets: */

proc export data=PROJ1.Organics_s8

  outfile='/home/u58783680/INFS 762 Data Warehousing/Project 1/organics_s10_export.csv'

  dbms=csv replace;

run;

proc export data=PROJ1.Organics_s9_Training

  outfile='/home/u58783680/INFS 762 Data Warehousing/Project 1/organics_s10_Training.csv'

  dbms=csv replace;

run;

proc export data=PROJ1.Organics_s9_Validation

  outfile='/home/u58783680/INFS 762 Data Warehousing/Project 1/organics_s10_Validation.csv'

  dbms=csv replace;

run;
```

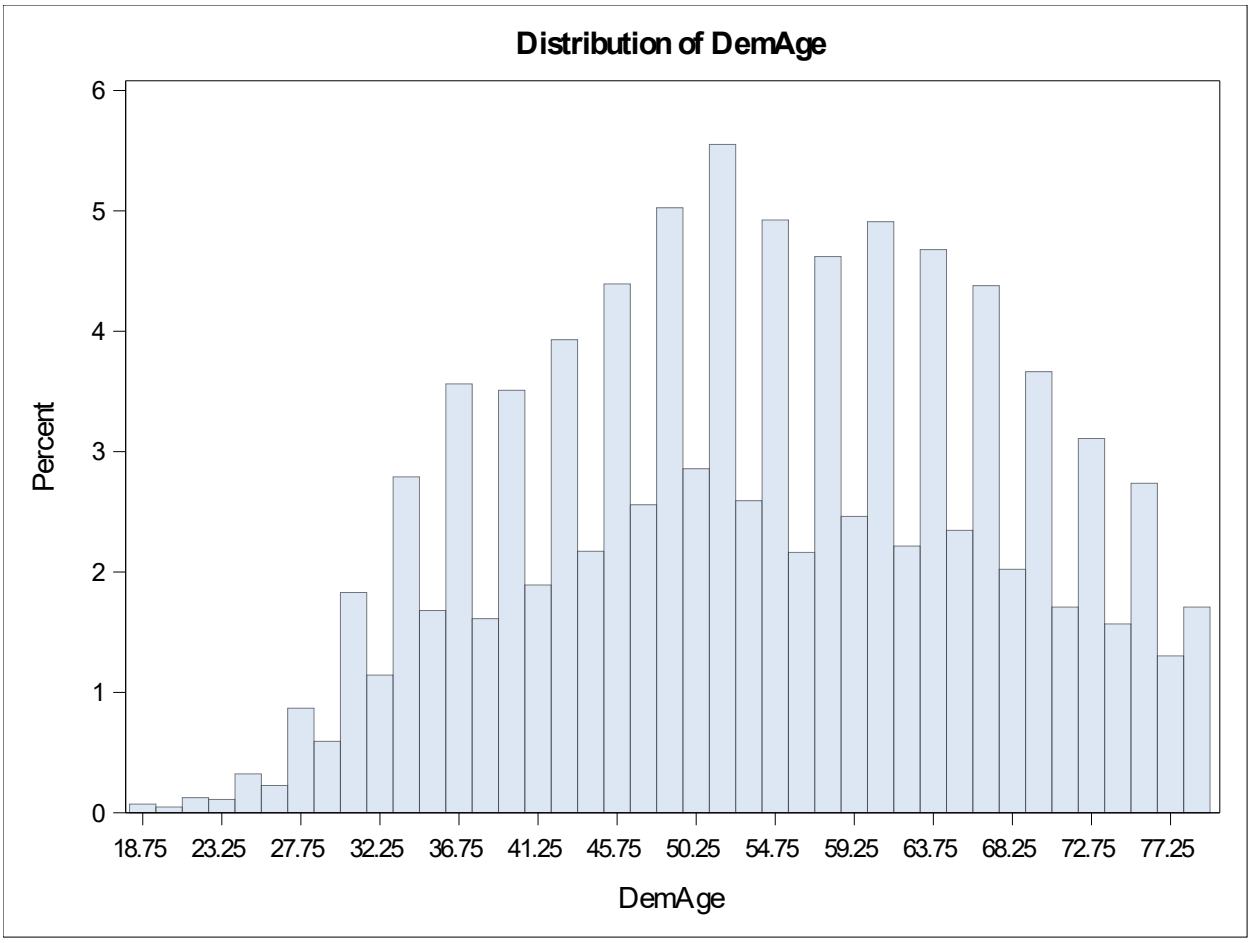
/* Fit models - Done in Weka */

2. Histograms of the continuous variables:




Distribution of DemAffI
Percent
12.5
10.0
7.5
5.0
2.5
0.0
0.4
2.8
5.2
7.6
10.0
12.4
14.8
17.2
19.6
22.0
24.4
26.8
29.2
31.6
34.0
DemAffI


Distribution of PromTime
Percent
20
15
10
5
0
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
PromTime



3.

Variables that have missing values:

- DemAffl
- DemAge
- PromTime
- DemClusterGroup
- DemGender
- DemReg
- DemTVReg

Variables that have skewed distributions:

- Promspend which is skewed to the right
- PromTime which is skewed to the right
- DemAffl which is very slightly skewed to the right

4.

Tell me which variables you selected in the first round of variable selection.

- DemAffl
- DemAge
- DemGender
- DemAffl_M
- DemAge_M
- DemGender_M

5.

Use the variables you selected in the previous step to do model fitting. Tell me which three models (including logistic regression) you used.

- Logistic Regression
- K-Nearest Neighbors (5)
- Random Forest

6.

Submit a table that shows the precision/recall/accuracy of the three models.

Models results (TargetBuy = dependent variable with 1 being the positive result)	Precision (rank)	Recall (rank)	Accuracy (rank)
Logistic Regression	71.6% (1st)	41.5% (3rd)	81.3% (1st)
K-Nearest Neighbors (5)	67.3% (2nd)	42.4% (2nd)	80.5% (2nd)
Random Forest	64.1% (3rd)	45.7% (1st)	80.1% (3rd)

Based on the above results, logistic regression ended up producing the best model since it had the highest precision and accuracy rates; although lower recall rates than Random Forest showing that it's worse at detecting the classes of TargetBuy through the class imbalance.

7.

After you transformed the variables with skewed distribution, please do a variable selection again and tell me which variables you selected. Fit the three models and submit a table that shows the precision/recall/accuracy of the three models.

Variables selected:

- DemAffl_Log
- DemAge

- DemGender
- DemAffl_M
- DemAge_M
- DemGender_M

Models results (TargetBuy = dependent variable with 1 being the positive result)	Precision (rank)	Recall (rank)	Accuracy (rank)
Logistic Regression	71.5% (1st)	42.1% (2nd)	81.4% (1st)
K-Nearest Neighbors (5)	67.3% (2nd)	42.1% (2nd)	80.4% (2nd)
Random Forest	63.9% (3rd)	45.9% (1st)	80.0% (3rd)

8.

Please tell if log transformation results in better model performance.

The log transformations results in roughly the same model performance, with slightly better performance with Logistic Regression and slightly worse performance with K-Nearest Neighbors and Random Forest. With the transformed and untransformed datasets, random forest does slightly better with detecting the classes despite the 3 to 1 ratio class imbalance for TargetBuy.

9.

Nothing else.