Before partitioning the dataset, we need to impute the missing values to build a model which is not biased. There are various ways to impute data such as rpart, mice and so on. I have gone with a simple approach of imputing empty values with mean for numeric data and max for categorical data.

sapply(organics, function(x) sum(is.na(x)))

organics$DemAffl <- with(organics, impute(DemAffl, mean))

organics$DemAge <- with(organics, impute(DemAge, mean))

organics$DemClusterGroup <- with(organics, impute(DemClusterGroup, max))

organics$DemGender <- with(organics, impute(DemGender, max))

organics$DemReg <- with(organics, impute(DemReg, max))

organics$DemTVReg <- with(organics, impute(DemTVReg, max))

organics$PromTime <- with(organics, impute(PromTime, mean))

The next step is to randomize the data by setting a seed of 42. Randomization is done to avoid bias in the results of the test data set. If the first 100 rows of a dataset contains similar data and if we base our model based on this data, our prediction for the test data would turn out wrong. Randomization eliminates this.

The dataset is partitioned 50/50 as mentioned

organicstrain <- organicsrand[1:11111, ]

organicstest <- organicsrand[11112:22223, ]

Distribution of Target Variable(After Randomization):

table(organics$TargetBuy)

0 1

16718 5505

> table(organicstrain$TargetBuy)

0 1

8395 2716

> table(organicstest$TargetBuy)

0 1

8323 2789

As we can see from the distribution in the training and test data set, the number of people who have purchased organics are very similar and almost equally distributed.

**Proportion of people who have purchased organic products** = (5505)/(22223) = 0.2477= 24.77%

The variable TargetAmt directly gives the number of organic products purchased by the customer. Any non-zero number in the TargetAmt column would mean that a particular consumer purchased an organic product which would render the other variables and hence the need to build a model insignificant.

**Hence, TargetAmt should not be used as an input to build the model.**

**Decision Tree- Training data set**

n= 11111

node), split, n, loss, yval, (yprob)

\* denotes terminal node

1) root 11111 2716 0 (0.7555576 0.2444424)

2) DemAge>=44.5 8364 1276 0 (0.8474414 0.1525586)

4) DemAffl< 13.5 7862 1021 0 (0.8701348 0.1298652) \*

5) DemAffl>=13.5 502 247 1 (0.4920319 0.5079681)

10) DemGender=,M,U 191 60 0 (0.6858639 0.3141361) \*

11) DemGender=F 311 116 1 (0.3729904 0.6270096) \*

3) DemAge< 44.5 2747 1307 1 (0.4757918 0.5242082)

6) DemGender=,M,U 1018 327 0 (0.6787819 0.3212181)

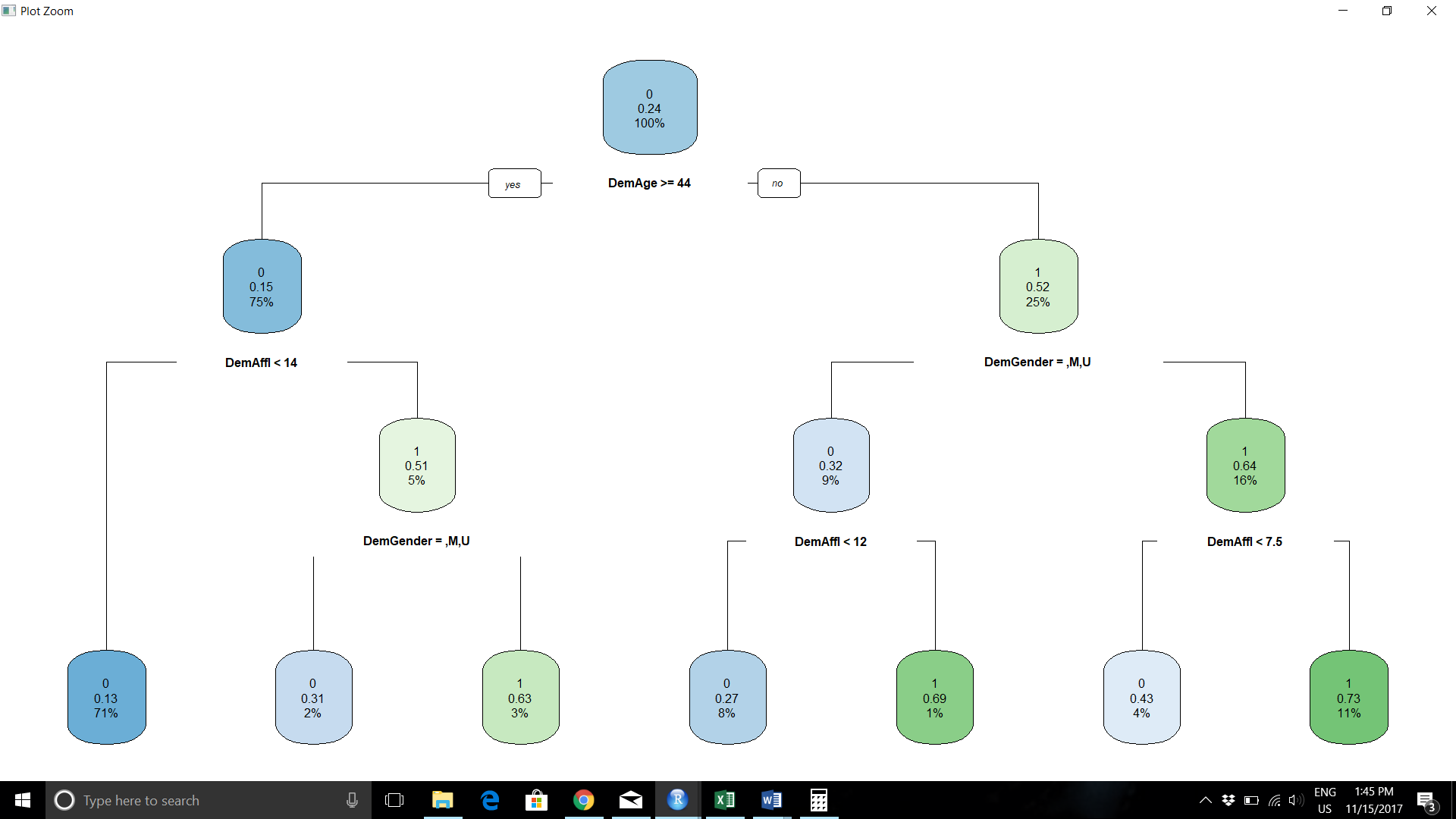
12) DemAffl< 12.5 883 234 0 (0.7349943 0.2650057) \*

13) DemAffl>=12.5 135 42 1 (0.3111111 0.6888889) \*

7) DemGender=F 1729 616 1 (0.3562753 0.6437247)

14) DemAffl< 7.5 489 210 0 (0.5705521 0.4294479) \*

15) DemAffl>=7.5 1240 337 1 (0.2717742 0.7282258) \*



In the Decision tree 0’s and 1’s represent the action of buying and not buying respectively.

The values 0.15, 0.52 and so on indicate the probability of 0’s and 1’s.

The percentage represent the proportion of 0’s and 1’s.

**DemGender, DemAffl and DemAge are the main variables based on which the decision tree is arrived**

**There are 7 leaves in the decision tree obtained from the training dataset. Leaves are entities where no more split can occur.**

**DemAge>=44 is the first node on which the split starts**.

**Pruning the tree**:

Pruning the tree did not change my decision tree in any way.

> printcp(organicstree)

Classification tree:

rpart(formula = TargetBuy ~ ., data = organicstrain, method = "class")

Variables actually used in tree construction:

[1] DemAffl DemAge DemGender

Root node error: 2716/11111 = 0.24444

n= 11111

CP nsplit rel error xerror xstd

1 0.091495 0 1.00000 1.00000 0.016679

2 0.025405 2 0.81701 0.81701 0.015516

3 0.018778 3 0.79161 0.81517 0.015503

4 0.014543 4 0.77283 0.79050 0.015324

5 0.010000 6 0.74374 0.77209 0.015186

The value of xe error which is used as a base to make a decision on pruning the tree is the lowest corresponding to the last row(max split). Hence, no pruning is required for the decision tree.

**Confusion matrix and accuracy for training data set:**

Predicted

Actual 0 1

0 7900 495

1 1525 1191

> trainaccuracy

[1] 0.8181982

**Confusion matrix and accuracy for test data set:**

Predicted

Actual 0 1

0 7805 518

1 1602 1187

> testaccuracy

[1] 0.8092153

Comparing the 2 accuracies it is safe to say that the training data model was a good enough base for the test data. However, this conclusion can be drawn only for a partition of 50-50 and we cannot generalize this conclusion for every proportion of training and test data set.

Instead of manually calculating the accuracy, which is **(true pos + true neg)/sum of all elements of matrix**, I used this snippet:

organicstrain$correct <- organicstrain$TargetBuy == organicstrain$pred

traincorrectcount <- length(which(organicstrain$correct))

trainincorrectcount <- nrow(organicstrain) - traincorrectcount

trainerrorrate <- trainincorrectcount/nrow(organicstrain)

trainaccuracy <- 1-trainerrorrate

trainaccuracy

**Logistic Regression**:

Output by running the glm function. The glm function gives a good description of the error distribution

Deviance Residuals:

Min 1Q Median 3Q Max

-2.1561 -0.6866 -0.4168 -0.1255 3.0817

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -1.723e+00 2.881e-01 -5.982 2.20e-09 \*\*\*

organicstrain$PromClassPlatinum -2.357e-01 2.062e-01 -1.143 0.25301

organicstrain$PromClassSilver 4.240e-02 7.791e-02 0.544 0.58627

organicstrain$PromClassTin 7.633e-03 8.771e-02 0.087 0.93065

organicstrain$PromSpend 3.260e-06 5.652e-06 0.577 0.56411

organicstrain$DemAffl 2.434e-01 8.446e-03 28.820 < 2e-16 \*\*\*

organicstrain$DemAge -5.317e-02 2.337e-03 -22.750 < 2e-16 \*\*\*

organicstrain$PromTime -7.813e-03 6.475e-03 -1.207 0.22753

organicstrain$DemClusterGroupA -2.913e-02 1.746e-01 -0.167 0.86748

organicstrain$DemClusterGroupB -5.777e-02 1.564e-01 -0.369 0.71190

organicstrain$DemClusterGroupC 1.359e-02 1.541e-01 0.088 0.92973

organicstrain$DemClusterGroupD -5.777e-02 1.540e-01 -0.375 0.70747

organicstrain$DemClusterGroupE 5.825e-02 1.609e-01 0.362 0.71739

organicstrain$DemClusterGroupF 1.977e-03 1.543e-01 0.013 0.98978

organicstrain$DemClusterGroupU 3.174e-01 4.431e-01 0.716 0.47371

organicstrain$DemGenderF 1.596e+00 1.083e-01 14.730 < 2e-16 \*\*\*

organicstrain$DemGenderM 5.731e-01 1.173e-01 4.887 1.03e-06 \*\*\*

organicstrain$DemGenderU -6.303e-01 1.924e-01 -3.277 0.00105 \*\*

organicstrain$DemRegMidlands 3.799e-05 1.805e-01 0.000 0.99983

organicstrain$DemRegNorth 1.418e-02 1.837e-01 0.077 0.93847

organicstrain$DemRegScottish -9.187e-02 2.040e-01 -0.450 0.65241

organicstrain$DemRegSouth East 1.395e-02 1.794e-01 0.078 0.93805

organicstrain$DemRegSouth West 1.443e-01 2.269e-01 0.636 0.52494

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 12358.7 on 11110 degrees of freedom

Residual deviance: 9600.5 on 11088 degrees of freedom

AIC: 9646.5

Number of Fisher Scoring iterations: 5

**It is very clear from the model that DemAffl, DemAge and DemGender are the most significant parameters to be considered**.

We take these dominating variables and run another model

Deviance Residuals:

Min 1Q Median 3Q Max

-2.1058 -0.6879 -0.4165 -0.1253 3.0532

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -1.721756 0.166122 -10.364 < 2e-16 \*\*\*

DemAffl 0.243398 0.008430 28.874 < 2e-16 \*\*\*

DemAge -0.053855 0.002112 -25.497 < 2e-16 \*\*\*

DemGenderF 1.598719 0.108223 14.772 < 2e-16 \*\*\*

DemGenderM 0.576759 0.117216 4.920 8.63e-07 \*\*\*

DemGenderU -0.626117 0.192291 -3.256 0.00113 \*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 12358.7 on 11110 degrees of freedom

Residual deviance: 9608.7 on 11105 degrees of freedom

AIC: 9620.7

Number of Fisher Scoring iterations: 5

The same number of iterations is obtained on choosing only the above mentioned 3 parameters.

Even the null deviance, residual deviance and AIC values are not that different between the 2 models.

**Build confidence intervals and calculate odds ratio**:

> confint.default(organicslogit)

2.5 % 97.5 %

(Intercept) -2.04734956 -1.39616249

DemAffl 0.22687633 0.25992028

DemAge -0.05799479 -0.04971499

DemGenderF 1.38660607 1.81083185

DemGenderM 0.34701940 0.80649834

DemGenderU -1.00300130 -0.24923300

Odds Ratio

> exp(coef(organicslogit))

(Intercept) DemAffl DemAge DemGenderF DemGenderM DemGenderU

0.1787520 1.2755766 0.9475696 4.9466915 1.7802590 0.5346638

Odds ratio= Prob of event occurring/prob of event not occurring

DemGenderF has the highest odds ratio which indicates that it has a significant effect on the model as odds ratio are a representation of the coefficients of the parameters.

**Confusion matrix and accuracy for training data**:

Predicted

Actual 0 1

0 7973 422

1 1653 1063

> trainaccuracylogit

[1] 0.8181982

**Confusion matrix and accuracy for test data:**

Predicted

Actual 0 1

0 7883 440

1 1703 1086

> testaccuracylogit

[1] 0.8132481

Comparing the 2 accuracies it is safe to say that the training data model was a good enough base for the test data. However, this conclusion can be drawn only for cut off value of 0.5 and the accuracy will change based on the cut-off value chosen.

Sensitivity = True Pos/(True Pos + False Neg)- also known as True positive rate. Higher value is desirable

**Accuracy of test data using decision tree: 80.92%**

**Sensitivity of test data using decision tree: 82.97%**

**Accuracy of test data using logistic regression: 80.92%**

**Sensitivity of test data using logistic regression: 82.23%**

**The sensitivity and accuracy of test data across the 2 models does not vary much and hence there is not much difference in the classification technique used in this case. However, there may be appreciable differences in the model based on the way we impute values to the cut off value to be chosen for a particular experiment.**

**For example, if MICE or RPart or RandomForest had been used to impute values or if the partition of the dataset had been set at a different proportion, we could have arrived at a completely different decision tree. Likewise, if we had used a different cut off value for the logistic regression instead of the default 0.5 we could have got different results for accuracy and sensitivity.**

**At the end of the day, the best model does not depend on the derived metrics alone but also on the method and the judgement call made for that particular model.**