R Notebook: Binary choice modeling

Packages

Make sure the following packages are installed before proceeding:

1. ggplot2
2. ggthemes
3. xtable
4. knitr
5. caret
6. e1071
7. pROC

```
library("xtable") # processing of regression output
library("knitr") # used for report compilation and table display
library("ggplot2") # very popular plotting library ggplot2
library("ggthemes") # themes for ggplot2
library("caret") # confusion matrix
```

```
## Loading required package: lattice
```

```
library("pROC") # confusion matrix
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

Binary choice modeling

This notebook shows how to estimate a simple binary choice model, interpret it, and use it to make predictions about consumer behavior.

Reading data

Let us load the data first.

```
RFMdata <- read.csv(file = "RFMData.csv",row.names=1)
kable(head(RFMdata,5),row.names = TRUE)
```

	Recency	Frequency	Monetary	Purchase
1	120	7	41.66	0
2	90	9	46.71	0
3	120	6	103.99	1
4	270	17	37.13	1
5	60	5	88.92	0

Each row (observation) is a separate customer who has transacted at least once before. The columns (variables) are:

1. Recency – how many days since last purchase
2. Frequency – how many times the consumer buys per year
3. Monetary – total $ amount spent per year
4. Purchase - (yes/no) whether purchase occurred

Naive model

Now, let us draw a scatter plot of purchase occurrences (y-axis) by recency (x-axis). We will also overlay on top a regression line through the cloud of points that is based on equation

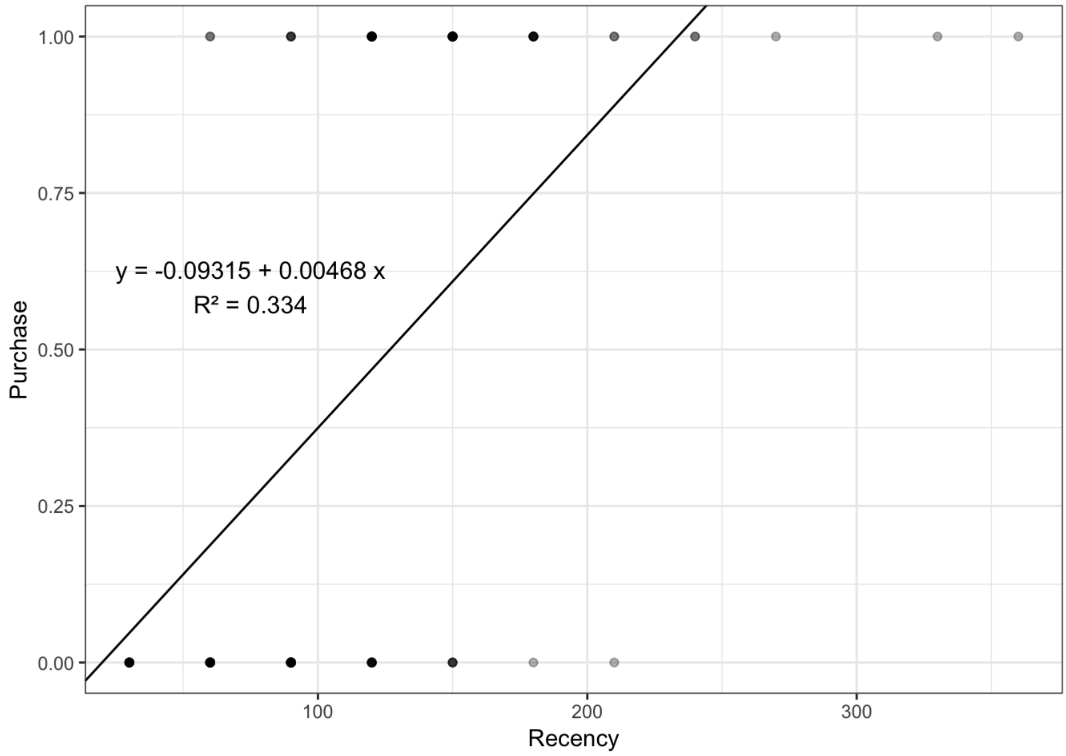
\[Purchase_i=\beta_0+\beta_1Recency_i\] We estimate parameters \(\beta_0,\beta_1\) using ordinary least squares – lm() function below. Then we plot everything using ggplot2 package, and use ggthemes to make the plot look nice.

```
model <- lm(data=RFMdata, Purchase ~ Recency) # note, lm() automatically includes intercept

# coef(model)[1] is beta0
# coef(model)[2] is beta1

p <- ggplot(RFMdata, aes(Recency, Purchase)) +
  geom_point(alpha=0.3) + # draws points
  theme_bw() # changes visual theme of the plot to make the look cleaner

p + geom_abline(intercept = coef(model)[1], # setting intercept of the line based on beta0
                slope = coef(model)[2]) + # setting slope of the line based on beta1
  # annotating
  annotate(label = sprintf("y = %.5f + %.5f x\nR² = %.3f", coef(model)[1],coef(model)[2],  summary(model)$r.
squared), geom = "text", x = 75, y = 0.6, size = 4)
```



What is naive about this model? For high values of recency (e.g., over 200), regression predicts values above 1, which is outside of the range of valid values.

A better choice model – Logit

A better model is logit, which restricts the output values to lie in \([0,1]\) interval.

Specifically, it expresses probability of a purchase by customer \(i\) as a function of coefficients \(\beta_{0:3}\) and variables in the following manner: \[P(Purchase_i) = \frac{\exp(\beta_0 + \beta_1Recency_i + \beta_2Frequency_i+\beta_3Monetary_i)}{\exp(\beta_0 + \beta_1Recency_i + \beta_2Frequency_i+\beta_3Monetary_i) + 1}\] Intuitively, utility of *choosing to buy* is \[V_{bi} = \beta_0 + \beta_1Recency_i + \beta_2Frequency_i+\beta_3Monetary_i\] whereas utility of *choosing **not** to buy* is normalized to zero \(V_{ni}=0\), so \(exp(V_{n})=exp(0)=1\) in the fraction above.

With the given formulation, we can estimate values \(\beta_{0:3}\) that fit data best. We use glm() of family="binomial".

```
model <- glm(Purchase~Recency+Frequency+Monetary, data=RFMdata, family = "binomial")
output <- cbind(coef(summary(model))[, 1:4],exp(coef(model)))
colnames(output) <- c("beta","SE","z val.","Pr(>|z|)",'exp(beta)')
kable(output,caption = "Logistic regression estimates")
```

Logistic regression estimates

	beta	SE	z val.	Pr(>|z|)	exp(beta)
(Intercept)	-30.2976692	8.5522913	-3.542638	0.0003961	0.000000
Recency	0.1114175	0.0309797	3.596464	0.0003226	1.117862
Frequency	0.5941268	0.2429393	2.445577	0.0144620	1.811448
Monetary	0.1677054	0.0465645	3.601572	0.0003163	1.182588

We also run the likelihood ratio test with $H_0:\beta_1=\beta_2=\beta_3=0$ – to make sure our full logit model offers a significantly better fit than the model with just an intercept. We find that $\chi^2=107.14$ and $P(>|Chi|)\approx 0$, so we reject H_0.

```
# likelihood ratio test
reduced.model <- glm(Purchase ~ 1, data=RFMdata, family = "binomial")
kable(xtable(anova(reduced.model, model, test = "Chisq")),caption = "Likelihood ratio test")
```

Likelihood ratio test

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
99	137.62776	NA	NA	NA
96	30.48715	3	107.1406	0

Predicting probabilities

Now we calculate $P(Purchase_i)$ for each individual in the data set.

```
# calculate logit probabilities
RFMdata$Base.Probability <- predict(model, RFMdata, type="response")
kable(head(RFMdata,5),row.names = TRUE)
```

	Recency	Frequency	Monetary	Purchase	Base.Probability
1	120	7	41.66	0	0.0030728
2	90	9	46.71	0	0.0008332
3	120	6	103.99	1	0.9833225
4	270	17	37.13	1	0.9999999
5	60	5	88.92	0	0.0032378

Predicting behavior

We also calculate an indicator variable for whether individuals will purchase or not, based on their predicted probabilities $\mathbb{1}[P(Purchase_i)\geq 0.5]$ If individual's predicted probability is greater or equal to 0.5, we predict he will make a purchase.

```
# purchase vs. no purchase <-> p>0.5 or p<0.5
RFMdata$Predicted.Purchase <- 1*(RFMdata$Base.Probability>=0.5)
kable(head(RFMdata,5),row.names = TRUE)
```

	Recency	Frequency	Monetary	Purchase	Base.Probability	Predicted.Purchase
1	120	7	41.66	0	0.0030728	0
2	90	9	46.71	0	0.0008332	0
3	120	6	103.99	1	0.9833225	1
4	270	17	37.13	1	0.9999999	1
5	60	5	88.92	0	0.0032378	0

Evaluating the model

Now we compute a *confusion matrix* between predicted purchases and actual purchase behavior.

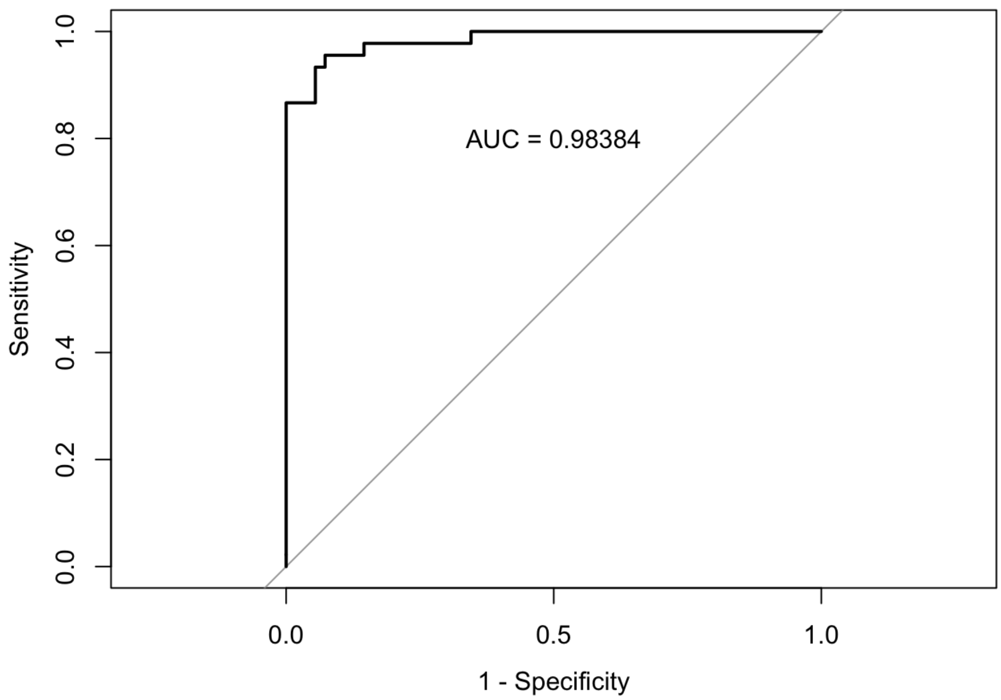
```
confusionMatrix(table(RFMdata$Predicted.Purchase,RFMdata$Purchase),positive = "1")
```

```
## Confusion Matrix and Statistics
## 
##    
##      0  1
##   0 51  2
##   1  4 43
##                                           
##                Accuracy : 0.94            
##                  95% CI : (0.874, 0.9777) 
##     No Information Rate : 0.55            
##     P-Value [Acc > NIR] : <2e-16          
##                                           
##                   Kappa : 0.8793          
##  Mcnemar's Test P-Value : 0.6831          
##                                           
##             Sensitivity : 0.9556          
##             Specificity : 0.9273          
##          Pos Pred Value : 0.9149          
##          Neg Pred Value : 0.9623          
##              Prevalence : 0.4500          
##          Detection Rate : 0.4300          
##    Detection Prevalence : 0.4700          
##       Balanced Accuracy : 0.9414          
##                                           
##        'Positive' Class : 1               
## 
```

We can also plot the receiver operating characteristic (ROC) curve, which illustrates the diagnostic ability of a binary logit model. It is created by plotting the true positive rate (TPR) against the false positive rate (FPR) – at various decision threshold values for prediction.

ROC curve can be quickly evaluated using area under the curve (AUC) metric, which captures the overall quality of the classifier. The greater the AUC, the better. AUC of 1.0 represents a perfect classifier, AUC of 0.5 (diagonal line) represents a worthless classifier. As we see, binary logit classifier does a good job predicting purchases on the training data.

```
rocobj <- roc(RFMdata$Purchase, RFMdata$Base.Probability)
{plot(rocobj,legacy.axes=TRUE)
text(0.5, 0.8, labels = sprintf("AUC = %.5f",rocobj$auc))}
```



Finally, we predict new probabilities under a hypothetical scenario that everyone's *Monetary* variable went up by one unit \[V_{bi}^{new} = \beta_0 + \beta_1Recency_i + \beta_2Frequency_i+\beta_3(Monetary_i+1)\]

```r
# calculate new logit probabilities (Monetary+1)
RFMdata_new <- RFMdata
RFMdata_new$Monetary <- RFMdata_new$Monetary + 1
RFMdata$New.Probability <- predict(model, RFMdata_new, type="response")
```

We compare mean new probability across individuals to the mean of old probabilities, and also calculate the lift metric.

$$p_{old}=\frac{1}{N}\sum_{i=1}^{N} P(Purchase_i) = \frac{1}{N}\sum_{i=1}^{N} \frac{\exp(V_{bi})}{\exp(V_{bi}) + 1}=\frac{1}{N}\sum_{i=1}^{N}\frac{\exp(\beta_0 + \beta_1Recency_i + \beta_2Frequency_i+\beta_3Monetary_i)}{\exp(\beta_0 + \beta_1Recency_i + \beta_2Frequency_i+\beta_3Monetary_i) + 1}$$

$$p_{new}=\frac{1}{N}\sum_{i=1}^{N} P(Purchase_i^{new}) = \frac{1}{N}\sum_{i=1}^{N} \frac{\exp(V_{bi}^{new})}{\exp(V_{bi}^{new}) + 1}=\frac{1}{N}\sum_{i=1}^{N}\frac{\exp(\beta_0 + \beta_1Recency_i + \beta_2Frequency_i+\beta_3(Monetary_i+1))}{\exp(\beta_0 + \beta_1Recency_i + \beta_2Frequency_i+\beta_3(Monetary_i+1)) + 1}$$

$$Lift = \frac{p_{new}-p_{old}}{p_{old}}$$

```r
# mean predicted base probability
mean(RFMdata$Base.Probability)
```

```
## [1] 0.45
```

```r
# mean new predicted probability
mean(RFMdata$New.Probability)
```

```
## [1] 0.4578851
```

```r
# lift
(mean(RFMdata$New.Probability) - mean(RFMdata$Base.Probability))/mean(RFMdata$Base.Probability)
```

```
## [1] 0.01752255
```

```r
# remove predicted purchase variable
RFMdata$Predicted.Purchase <- NULL

# data
kable(head(RFMdata,5),row.names = TRUE)
```

	Recency	Frequency	Monetary	Purchase	Base.Probability	New.Probability
1	120	7	41.66	0	0.0030728	0.0036319
2	90	9	46.71	0	0.0008332	0.0009852
3	120	6	103.99	1	0.9833225	0.9858611
4	270	17	37.13	1	0.9999999	0.9999999
5	60	5	88.92	0	0.0032378	0.0038267