実証サンプル課題

Kei Sakamoto

2019/12/7

データの読み込み。

今回はDownloadフォルダにダウンロードし、そこからRで読み込むとします。

```
library(readr)
user_visit <- read_csv("Downloads/user_visit.csv")
```

```
## Parsed with column specification:
## cols(
##   user_id = col_character(),
##   referrer = col_character(),
##   timestamp = col_datetime(format = "")
## )
```

```
user_purchase <- read_csv("Downloads/user_purchase.csv")
```

```
## Parsed with column specification:
## cols(
##   user_id = col_character(),
##   item_id = col_character(),
##   price = col_double(),
##   timestamp = col_datetime(format = "")
## )
```

データクリーニング

方向性としては、欠損や9月のデータ、2回以上購入したuserのuser_idの重複を消し、非購入購入を表す(0,1)変数を１つのカラムとして作ります。

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.2.1 ──
```

```
## ✔ ggplot2 3.1.0     ✔ purrr   0.3.2
## ✔ tibble  2.1.1     ✔ dplyr   0.8.0.1
## ✔ tidyr   0.8.3     ✔ stringr 1.4.0
## ✔ ggplot2 3.1.0     ✔ forcats 0.4.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
# user_purchaseのpriceカラムがNAでないということは何かしら買ったということで、金額は区別せず買ったらpriceを1に置き換えます。
user_purchase$price[!is.na(user_purchase$price)] <- 1
# 逆にNAの場合は"欠損"という値を入れておきます。後々user_visitと合わせる時に買っていない場合のNAと欠損データの場合を区別するためです。
user_purchase$price[is.na(user_purchase$price)] <- "欠損"
#キーをuser_idとしてuser_idが被っていれば列のみの結合、被っていなければさらに行ごと追加して結合してuser_infoという統合されたデータを作ります。
user_info <- full_join(user_purchase, user_visit, by = "user_id")
user_info2 <- user_info #補足の考察の時に使います。
#購入の有無だけに焦点を当てる場合、一人で二回以上購入した場合に記録されるuser_id の重複はいらないので削ります。
user_info <- user_info %>% distinct(user_id, .keep_all=TRUE)
#10月1日以外のデータは分析の対象外なので削ります。
user_info <- subset(user_info, grepl("2018-10-01", user_info$timestamp.y))
#欠損は非購入とは区別して今回は消すこととします。user_purchaseのデータにNAがあり、item_idとpriceは必要ないので買ったとみなすこともできなくはないですが、timestampまでないとなると何らかのエラーの可能性も考え、さらに十分にサンプル数も多いのでそのような対応にしました。
user_info <- subset(user_info, is.na(price)| price != "欠損" )
#非購入者のpriceをNAから0に置き換えます。
user_info$price[is.na(user_info$price)] <- 0
#先ほど"欠損"をpriceに入れた影響でclassがcharacterになっているので、回帰分析で使いやすいようにnumericに変換します。
user_info$price <- as.numeric(user_info$price)
#referrerも同様にcharacterからfactorに変換しておくことで一部回帰分析に用いやすくなります。
user_info$referrer<-as.factor(user_info$referrer)
#item_idも無駄に消す必要はないですが、同時に今回の題材に対しては残す必要もないので見やすさの為にuser_infoにおいては消しておきます。
user_info <- user_info %>% select(-item_id)
#timestampも同様に消す必要も残す必要もないですがイメージのために訪れた時間と購入時間に名前を変えておきます。購入費購入で1かNAが割り当てられてきたpriceカラムはbuyという名前に変更します。
user_info <- rename(user_info, buy = "price", bought_at = "timestamp.x", visited_at = "timestamp.y")
```

欠損やuser_idの被りのないデータでの(経路別の)購入の有無に関する概要

まずは全体、つまり10月1日にキャンペーンサイトに訪れた人数とそのうちの購入者の割合を確認

```
nrow(subset(user_info, buy ==1))/nrow(user_info) #434/1744 = 0.2488532
```

```
## [1] 0.2488532
```

ウェブ広告からの場合

```
via_ad <- subset(user_info,referrer == "ad")
n_via_ad <-nrow(via_ad) #355 nはnumberの略の意図があります。
n_bought_via_ad <- nrow(subset(via_ad, buy == 1)) #31
n_bought_via_ad/n_via_ad #31/355 = 0.08732394
```

```
## [1] 0.08732394
```

ウェブ検索からの場合

```
via_search <- subset(user_info,referrer == "search")
n_via_search <- nrow(via_search) #482
n_bought_via_search <- nrow(subset(via_search, buy ==1)) #169
n_bought_via_search/n_via_search #169/482 = 0.3506224
```

```
## [1] 0.3506224
```

スマホアプリからの場合

```
via_app <- subset(user_info,referrer == "app")
n_via_app <- nrow(via_app) #907
n_bought_via_app <- nrow(subset(via_app, buy ==1)) #234
n_bought_via_app/n_via_app #234/907 = 0.2579934
```

```
## [1] 0.2579934
```

ウェブ検索からが一番購入割合が高いことがわかります。

このままtテストなどを実行しても良いのですが、 今後説明変数を追加することも考慮に入れて、線形確率(Linear Probability)モデルに拡張します。

母集団回帰直線のモデルは以下で考えます。

$$buy_i = \beta_0 + \beta_1 referrer_i + u_i$$

このモデルのβ_0,β_1をOLS(最小二乗法)で推定し、その推定量を$\hat{\beta}_0$,$\hat{\beta}_{1i}$というようにハット付きで置きます。

つまりfitted value(当てはめ値)の集合は

$$\hat{buy}_i = \hat{\beta}_0 + \hat{\beta}_{1i} referrer_i$$

で表せることになります。 非説明変数がbinary(0,1変数)の時、OLSの回帰直線に当たるE(buy_i)(= $\hat{buy}_i$)は説明変数が与えられた時buy_iが1になる条件つき確率に一致します。 今回はOLS推定量が一致性を保つための仮定は満たされているものとして進めます。

OLS

```
lpmreg<- lm(buy ~ referrer, data = user_info) #Linear Probability Model regression
summary(lpmreg)
```

```
##
## Call:
## lm(formula = buy ~ referrer, data = user_info)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35062 -0.25799 -0.25799 -0.08732  0.91268
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.08732    0.02246   3.889 0.000105 ***
## referrerapp    0.17067    0.02649   6.443 1.51e-10 ***
## referrersearch  0.26330    0.02959   8.898  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4231 on 1741 degrees of freedom
## Multiple R-squared:  0.04396,    Adjusted R-squared:  0.04286
## F-statistic: 40.03 on 2 and 1741 DF,  p-value: < 2.2e-16
```

```
#デフォルトではad がbase categoryとされていますが、以下のように他のカテゴリー(例えばapp)に置き直すこともできます。いずれも本質的には同じ結果です。
#user_info$referrer <- relevel(user_info$referrer, "app")
#lpmreg2<- lm(buy ~ referrer,data = user_info)
#summary(lpmreg2)
```

推定結果によると $\hat{\beta}_0 = 0.08732$、

$$\hat{\beta}_{1i} = \begin{cases} 0 & (referrer_i = "\ ad\ ") \\ 0.17067 & (referrer_i = "\ app\ ") \\ 0.26330 & (referrer_i = "\ search\ ") \end{cases}$$

$\hat{\beta}_{1i}$の解釈は「referrerがウェブ広告場合の購入/非購入割合と他のそれぞれの場合の購入/非購入割合の差」になります。

referrerがウェブ広告だった場合の購入/非購入比率は約8.7%、スマホアプリの場合は0.08732+0.17067から約25.8%、ウェブ検索の場合は0.08732+0.26330から約0.35.1%と解釈できます(少数第一位で丸めています)。

referrerがそれぞれの場合の購入確率の予測

```
cvalues <- list(referrer = c("ad","app","search"))
predict(lpmreg, cvalues)
```

```
##         1         2         3
## 0.08732394 0.25799338 0.35062241
```

仮説検定

線形確率モデルは誤差項は構造上均一分散になり得ないので、検定の時はsummaryでの結果をそのまま解釈はせず不均一分散を用います。

まず、それぞれの係数が0、つまりreferrerがad だった場合とそれぞれの場合で購入確率が変わるのではないかという説について統計的に検証します。

$$H_0(\text{帰無仮説}) : \hat{\beta}_{1i}(app) = 0, H_1(\text{対立仮説}) : \hat{\beta}_{1i}(app) \neq 0$$

$$H_0 : \hat{\beta}_{1i}(search) = 0, H_1 : \hat{\beta}_{1i}(search) \neq 0$$

のそれぞれの仮説検定を行います。referrerがadの時より他の方が高いということの根拠、先行研究などはないものとして両側検定にします。

```
library(lmtest);library(car)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
coeftest(lpmreg,vcov=hccm)
```

```
##
## t test of coefficients:
##
##                Estimate Std. Error t value  Pr(>|t|)
## (Intercept)    0.087324   0.015026  5.8116 7.342e-09 ***
## referrerapp    0.170669   0.020912  8.1614 6.279e-16 ***
## referrersearch 0.263298   0.026460  9.9509 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

これによってそれぞれの帰無仮説は有意水準0.1%で棄却されます。

さらに

$$H_0 : \hat{\beta}_{1i}(app) = \hat{\beta}_{1i}(search) = 0, H_1 : \text{少なくともどちらか一方は0でない}$$

の仮説検定、つまりそれぞれの係数が同時に0、つまりappからもsearchからの場合もどちらもadからの場合の購入確率とは変わらないという帰無仮説の検定を実行します。

```
lpmrest<- lm(buy ~ 1, data = user_info)
waldtest(lpmrest,lpmreg, vcov =hccm )
```

```
## Wald test
##
## Model 1: buy ~ 1
## Model 2: buy ~ referrer
##   Res.Df Df    F    Pr(>F)
## 1   1743
## 2   1741  2 59.6 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

「差がない」という帰無仮説は有意水準0.1%で棄却されます。

また、全く同じモデルは次のようにも表せます。

$$buy_i = \beta_0 + \beta_1 app_i + \beta_2 search_i + u_i$$

多重共線性の問題回避のため、ad_iという説明変数は落としています。

$$app_i = \begin{cases} 1 & (referrer_i = " \ app \ ") \\ 0 & (otherwise) \end{cases}$$

$$search_i = \begin{cases} 1 & (referrer_i = " \ search \ ") \\ 0 & (otherwise) \end{cases}$$

$$\hat{buy}_i = \hat{\beta}_0 + \hat{\beta}_1 app_i + \hat{\beta}_2 search_i$$

このモデルを作る為の列の追加

```
user_info<-transform(user_info,app=1)
user_info$app[user_info$referrer != "app" ] <- 0
user_info$app <- as.numeric(user_info$app)

user_info<-transform(user_info,search=1)
user_info$search[user_info$referrer != "search" ] <- 0
user_info$serach <- as.numeric(user_info$search)

user_info<-transform(user_info,ad=1)
user_info$ad[user_info$referrer != "ad" ] <- 0
user_info$ad <- as.numeric(user_info$ad)
```

本質的には同じOLS推定

```
lpmreg3 <-lm(buy~ app + search, data = user_info)
summary(lpmreg3)
```

```
##
## Call:
## lm(formula = buy ~ app + search, data = user_info)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.35062 -0.25799 -0.25799 -0.08732  0.91268
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.08732    0.02246   3.889 0.000105 ***
## app          0.17067    0.02649   6.443 1.51e-10 ***
## search       0.26330    0.02959   8.898  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4231 on 1741 degrees of freedom
## Multiple R-squared:  0.04396,    Adjusted R-squared:  0.04286
## F-statistic: 40.03 on 2 and 1741 DF,  p-value: < 2.2e-16
```

推定結果から

$$\hat{buy}_i = 0.08732 + 0.17067 app_i + 0.26330 search_i$$

が導き出されました。

“app”の係数と“search”の係数が同じであることを帰無仮説とするFテスト

$$H_0 : \hat{\beta}_1 = \hat{\beta}_2, H_1 : \hat{\beta}_1 \neq \hat{\beta}_2$$

```
myH0 <- c("app = search")
linearHypothesis(lpmreg3,myH0,vcov =hccm)
```

```
## Linear hypothesis test
##
## Hypothesis:
## app - search = 0
##
## Model 1: restricted model
## Model 2: buy ~ app + search
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df     F    Pr(>F)
## 1   1742
## 2   1741  1 12.51 0.0004154 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ここから、帰無仮説は有意水準0.1%でも棄却されるので、スマホアプリから来た場合とウェブ検索からきた場合も購入確率に差があると言えます。

Probit やLogit モデルにも拡張

将来的にさらに説明変数を追加する時、線形確率モデルだとpredictionが0から1の間に収まらないことがあるという潜在的な欠陥(確率の予測が0から1に収まらないのはおかしい)があるため、この拡張も用意しておきます。パラメータの推定方法は最尤法です。

Probit model

$$buy_i = \Phi(\beta_0 + \beta_1 app_i + \beta_2 search_i)$$

```
probitres<-glm(buy ~ app + search,family = binomial(link = "probit"), data = user_info)
#likelihood ratio test。ここからやはりadの時と購入確率には差があるといえます。
lrtest(probitres)
```

```
## Likelihood ratio test
##
## Model 1: buy ~ app + search
## Model 2: buy ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   3 -935.29
## 2   1 -978.51 -2 86.436  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#referrerがappの場合とsearchの場合で購入確率を予測します。
xpred<-list(app = c(1,0), search = c(0,1))
predict(probitres, xpred, type = "response")
```

```
##         1         2
## 0.2579934 0.3506224
```

Logit model

$$buy_i = \frac{1}{1 + \exp^{-}(\beta_0 + \beta_1 app_i + \beta_2 search_i)}$$

```
logitres<-glm(buy ~ app + search,family = binomial(link = "logit"), data = user_info)
#likelihood ratio test。ここからやはりadの時と購入確率には差があるといえます。
lrtest(logitres)
```

```
## Likelihood ratio test
##
## Model 1: buy ~ app + search
## Model 2: buy ~ 1
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   3 -935.29
## 2   1 -978.51 -2 86.436  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#同様にreferrerがappの場合とsearchの場合で購入確率を予測します。
xpred<-list(app = c(1,0), search = c(0,1))
predict(logitres, xpred, type = "response")
```

```
##         1         2
## 0.2579934 0.3506224
```

線形確率モデル、Probitモデル、Logitモデルそれぞれで同様の結果になります。

補足の考察

```
user_info2 <- subset(user_info2, grepl("2018-10-01", user_info2$timestamp.y))
user_info2 <- subset(user_info2, is.na(price)| price != "欠損" )

adset<-subset(user_info2,referrer == "ad")
nrow(subset(adset, item_id == "item01")) #例えばこのようにしてitem10まで購入件数を調べます。
```

```
## [1] 11
```

```
appset<-subset(user_info2,referrer == "app")
nrow(subset(appset, item_id == "item06"))#同様にしてitem10まで購入件数を調べます。
```

```
## [1] 11
```

```
searchset<-subset(user_info2,referrer == "search")
nrow(subset(searchset, item_id == "item06"))#同様にしてitem10まで購入件数を調べます。
```

```
## [1] 0
```

9月30日も含めた全訪問者数:1906

10月1日の訪問者数:1744(うちadから:355,appから:907,searchから:482)

10月1日の購入件数:877(うちadから:59,appから:493,searchから:325)

10月1日の購入者数:434(うちadから:31,appから:234,searchから:169)

10月1日のitem01購入件数:168

10月1日のitem02購入件数:164

10月1日のitem03購入件数:178

10月1日のitem04購入件数:148

10月1日のitem05購入件数:169

10月1日のitem06購入件数:11(うちappからが100%)

10月1日のitem07購入件数:8(うちappからが100%)

10月1日のitem08購入件数:9(うちappからが100%)

10月1日のitem09購入件数:10(うちappからが100%)

10月1日のitem10購入件数:12(うちappからが100%)

高額なitem06~item10(計50点)を買っているのは100%スマホアプリからの流入者だということがわかりました。

訪問者数に対する売上: ad:355人に対し5900, app:907人に対し54300(443×100+50×200), search:482人に対し32500

CVRだけでなく売り上げを考慮したとしてもsearchからが一番訪問者数に対する一人当たりの売上は高いです。

但し今は売上のみしかわかりません。 広告やアプリ導線設置のコスト、それぞれのitemのコストにもよりますが、例えばitem01からitem05が薄利多売型だとすれば、高額なitem06からitem10を買わせたい動機はより強まります。その場合アプリ導線の強化やスマホアプリからくるような人にもっと200の商品を買ってもらうような工夫が望まれます。利益率がどれも一緒なのであれば、やはりウェブ検索の経路をより強化することが良いと考えられます。また、仮に違うのは経路のみでキャンペーンサイトの内容や導線の踏ませ方など他の条件がが全て同じだとするなら、searchからの人に200の商品を買ってもらうようにするには本質的に商品の性質を変えることが必要と考えます。もしくは、1件もないというのは決定的な原因があるはずなのでsearchからのにで高額商品を買ってくれない原因を取り除くことが効果的だと考えられます。