**项目概况**

本项目的数据分析对象是天池大数据平台的竞赛《**O2O优惠券使用预测**》，要求根据用户在2016年1月1日至2016年6月30日之间的消费行为，预测用户在2016年7月领取优惠券后15天以内的使用情况。

平台提供的数据包括“用户线下消费和优惠券领取行为” 和“用户线上点击/消费和优惠券领取行为”，本项目仅使用用户线下数据进行分析和预测，该数据集共包括7个字段，如下：



1. **数据导入与查看**

> #设置工作空间载入包

> setwd("d:/data\_exercise/o2o")

> library(ggplot2)

> library(dplyr)

> library(InformationValue)

> library(parallel)

> #数据导入和查看

> train<-read.table(file = "ccf\_offline\_stage1\_train.csv",header = TRUE,sep = ",",stringsAsFactors = FALSE,na.strings = "null")

> test<-read.table(file="ccf\_offline\_stage1\_test\_revised.csv",header = TRUE,sep = ",",stringsAsFactors = FALSE,na.strings = "null")

> str(train)#查看数据结构

'data.frame': 1754884 obs. of 7 variables:

$ User\_id : int 1439408 1439408 1439408 1439408 1439408 1439408 1439408 1832624 2029232 2029232 ...

$ Merchant\_id : int 2632 4663 2632 2632 2632 2632 2632 3381 3381 450 ...

$ Coupon\_id : int NA 11002 8591 1078 8591 NA 8591 7610 11951 1532 ...

$ Discount\_rate: chr NA "150:20" "20:1" "20:1" ...

$ Distance : int 0 1 0 0 0 0 0 0 1 0 ...

$ Date\_received: int NA 20160528 20160217 20160319 20160613 NA 20160516 20160429 20160129 20160530 ...

$ Date : int 20160217 NA NA NA NA 20160516 20160613 NA NA NA ...

查看1-6月训练集中的所有优惠券领取和消费记录，总有175万条数据，每条数据包含7个变量。

对训练集和测试集进行重命名，将训练集的前6行显示如下：

> head(train)#显示训练集前六行

user shop coupon discount distance date\_r date\_u

1 1439408 2632 NA <NA> 0 NA 20160217

2 1439408 4663 11002 150:20 1 20160528 NA

3 1439408 2632 8591 20:1 0 20160217 NA

4 1439408 2632 1078 20:1 0 20160319 NA

5 1439408 2632 8591 20:1 0 20160613 NA

6 1439408 2632 NA <NA> 0 NA 20160516

总览数据概况，发现优惠券代码有较多行的缺失，没有消费券代码说明是线下直接消费的。最后一行消费日期也出现了大量缺失值，说明有相当比例的优惠券领取后并没有被使用。

> summary(train)#总览数据概况

user shop coupon discount distance

Min. : 4 Min. : 1 Min. : 1 Length:1754884 Min. : 0.00

1st Qu.:1845052 1st Qu.:1983 1st Qu.: 2840 Class :character 1st Qu.: 0.00

Median :3694446 Median :3532 Median : 7430 Mode :character Median : 0.00

Mean :3689255 Mean :4039 Mean : 6815 Mean : 2.36

3rd Qu.:5528759 3rd Qu.:6329 3rd Qu.:10323 3rd Qu.: 3.00

Max. :7361032 Max. :8856 Max. :14045 Max. :10.00

NA's :701602 NA's :106003

date\_r date\_u

Min. :20160101 Min. :20160101

1st Qu.:20160129 1st Qu.:20160306

Median :20160321 Median :20160416

Mean :20160316 Mean :20160399

3rd Qu.:20160514 3rd Qu.:20160523

Max. :20160615 Max. :20160630

NA's :701602 NA's :977900

下面进一步来计算消费券的实际领取和使用情况。

> #统计优惠券使用情况

> #没领券直接消费

> sum(is.na(train$coupon))/nrow(train)

[1] 0.3997996

> #领券，消费

> sum(!is.na(train$coupon)&!is.na(train$date\_u))/nrow(train)

[1] 0.04295555

> #领券，没消费

> sum(!is.na(train$coupon)&is.na(train$date\_u))/nrow(train)

[1] 0.5572448

> #用券消费占所有消费比例

> sum(!is.na(train$coupon)&!is.na(train$date\_u))/sum(!is.na(train$date\_u))

[1] 0.09701873

> #优惠券被使用的概率

> sum(!is.na(train$coupon)&!is.na(train$date\_u))/sum(!is.na(train$coupon))

[1] 0.07156868

可以看到所有数据记录中，直接消费占40%，有消费券领取信息的占60%。

进一步来看，在所有的消费中使用优惠券消费的占比为9.7%,可以看出大多数线下消费是没有用到优惠券的。

另一方面，计算所有领取的消费券的使用率为7%，表明绝大部分消费券领取后都未被使用。这里我们得出两点结论：

1. 大部分人线下消费是不会使用优惠券，以直接消费为主；
2. 大部分领取的优惠券没有在线下消费时使用。
3. **特征工程**

正式开始特征工程，从训练集中提取出有优惠券信息的子集。

> #提取有优惠券信息的子集

> train\_1<-filter(train,!is.na(train$discount))

**2.1提取优惠券是否在15日内使用的特征**

> #计算优惠券的使用时间

> train\_1$date\_r<-as.Date(as.character(train\_1$date\_r),"%Y%m%d")#转换领券日期格式

> train\_1$date\_u<-as.Date(as.character(train\_1$date\_u),"%Y%m%d")#转换消费日期格式

> train\_1$time<-difftime(train\_1$date\_u,train\_1$date\_r,units="days")

> train\_1$time<-as.integer(as.character(train\_1$time))

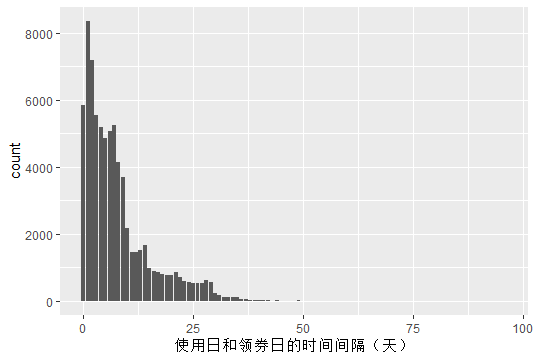
> #查看使用的优惠券的领取天数分布

> ggplot(train\_1,aes(x=train\_1$time))+

+ geom\_bar()+

+ labs(x="使用日和领券日的时间间隔（天）")

计算所有用掉的优惠券的使用和领取日的时间差，做出所有使用的时间分布如下，可以看到大部分使用的优惠券在领取15日以内消费，15日以上消费的占比很低。



下面进行特征创建，15日以内被使用为1，未使用或者15日以上使用为0。

> #定义函数，判断优惠券是否在15天内使用,0表示没使用，1表示使用

> use<-function(x)

+ if (is.na(x)|x>15) {return(0)}else

+ return(1)

> #增加特征

> train\_1$use<-sapply(train\_1$time,use)

> train\_1<-select(train\_1,user,shop,coupon,use,distance,date\_r,date\_u)#删除time列

查看正负样本比例，很清晰地，正样本共有 64395 例，负样本共有 988887 例。正负样本数量差别很大。这也是为什么会使用 AUC 作为模型性能评估标准的原因。

> table(train\_1$use)#看看正负样本比例

0 1

988887 64395

**2.2消费券打折特征提取**

> #查看打折率的内容

> unique(train\_1$discount)

[1] NA "150:20" "20:1" "200:20" "30:5" "50:10" "10:5"

[8] "100:10" "200:30" "20:5" "30:10" "50:5" "150:10" "100:30"

[15] "200:50" "100:50" "300:30" "50:20" "0.9" "10:1" "30:1"

[22] "0.95" "100:5" "5:1" "100:20" "0.8" "50:1" "200:10"

[29] "300:20" "100:1" "150:30" "300:50" "20:10" "0.85" "0.6"

[36] "150:50" "0.75" "0.5" "200:5" "0.7" "30:20" "300:10"

[43] "0.2" "50:30" "200:100" "150:5"

观察打折率这一列参数，我们可以知道优惠券的优惠方式分为直接打折、满减两种。不同优惠券的折扣率和使用门槛也有差异。

下面构建打折方式、打折率和使用门槛三个特征。

> #增加优惠券打折方式，折扣率，使用门槛特征

> discounttype<-function(x)#打折方式函数，无打折为NA,满减为1,直接打折为0

+ if (is.na(x)) {return(NA)}else

+ if (grepl(":",x)) {return(1)}else

+ return(0)

>

> discountrate<-function(x)#打折率函数

+ if (is.na(x)) {return(1)}else

+ if (grepl(":",x))

+ {a<-strsplit(x,":")

+ return(1-as.numeric(unlist(a)[2])/as.numeric(unlist(a)[1]))}else

+ return(as.numeric(x))

>

> man<-function(x)#满多少函数

+ if (grepl(":",x))

+ {return(as.numeric(unlist(strsplit(x,":"))[1]))

+ }else return(0)

>

> #增加三列特征

> cl <- makeCluster(6)

> train\_1$type<-parSapply(cl,train\_1$discount,discounttype)

> train\_1$rate<-parSapply(cl,train\_1$discount,discountrate)

> train\_1$man<-parSapply(cl,train\_1$discount,man)

处理之后我们可以看到训练集中多了type（打折类型）,rate（打折率）,man（使用门槛）三个新特征

> train\_1<-train\_1[,-5]

> head(train\_1)

user shop coupon use distance date\_r date\_u type rate man

1 1439408 4663 11002 0 1 2016-05-28 <NA> 1 0.8666667 150

2 1439408 2632 8591 0 0 2016-02-17 <NA> 1 0.9500000 20

3 1439408 2632 1078 0 0 2016-03-19 <NA> 1 0.9500000 20

4 1439408 2632 8591 0 0 2016-06-13 <NA> 1 0.9500000 20

5 1439408 2632 8591 0 0 2016-05-16 2016-06-13 1 0.9500000 20

6 1832624 3381 7610 0 0 2016-04-29 <NA> 1 0.9000000 200

下面对创建的三个特征进行分析。

（1）首先来看优惠券类型的影响。

> #1优惠券类型的影响

> ggplot(train\_1,aes(x=factor(train\_1$type),fill=factor(train\_1$use)))+#对比两种优惠券的使用率

+ geom\_bar(position = "dodge")+

+ geom\_text(stat = "count",aes(label=..count..),

+ position = position\_dodge(width = 1),vjust=-0.5)+

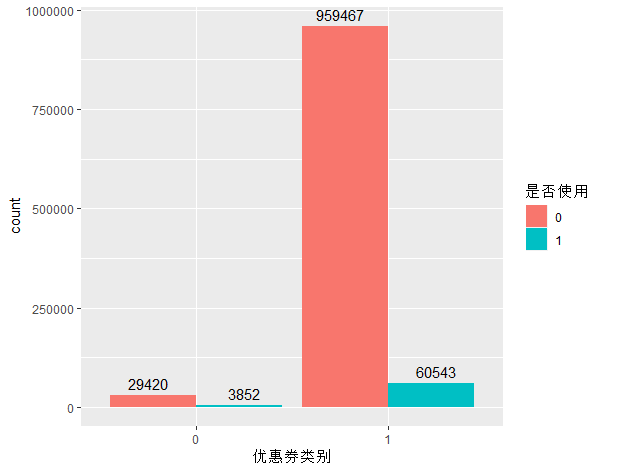
+ labs(x="优惠券类别",fill="是否使用")

> prop.table(table(train\_1$type,train\_1$use),1)#计算两类优惠券使用率

0 1

0 0.8842270 0.1157730

1 0.9406447 0.0593553



从上图可以看到满减优惠券占所有优惠券的绝大部分。计算得到满减优惠券的使用率为5.9%,直接打折优惠券的使用率为11.6%。一方面，两类优惠券的使用率均不高，另一方面，直接打折优惠券的使用率是满减类型的两倍。这里可以推断，优惠券的类型对使用率有一定影响。

> IV(X=factor(train\_1$type),Y=train\_1$use)#计算信息价值

[1] 0.02194767

attr(,"howgood")

[1] "Not Predictive"

计算信息价值，显示为"Not Predictive"

（2）下面来看折扣率的影响

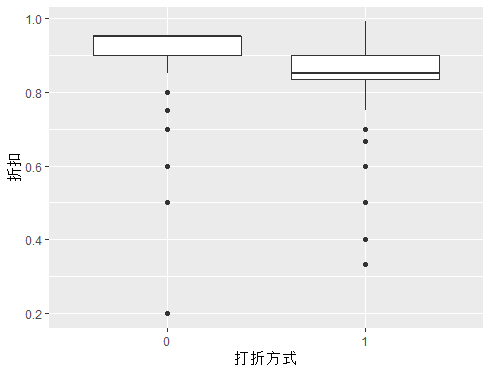
> #对比两种优惠券的折扣率

> ggplot(train\_1,aes(x=factor(train\_1$type),y=train\_1$rate))+

+ geom\_boxplot()+

+ labs(x="优惠券类别",y="折扣")

对比两类优惠券的折扣率，可以看到直接打折的折扣在主要在9-95折之间，而满减的基本在8-9折，满减的打折力度明显高于直接折扣。



> #对比不同折扣的使用率

> rate\_cut<-cut(train\_1$rate,breaks=c(0,0.5,0.7,0.8,0.9,1),labels = c("<0.5","0.5-0.7","0.7-0.8","0.8-0.9",">0.9"))

> ggplot(train\_1,aes(x=rate\_cut,fill=factor(train\_1$use)))+

+ geom\_bar(position = "dodge")+

+ labs(x="打折率",fill="是否使用")

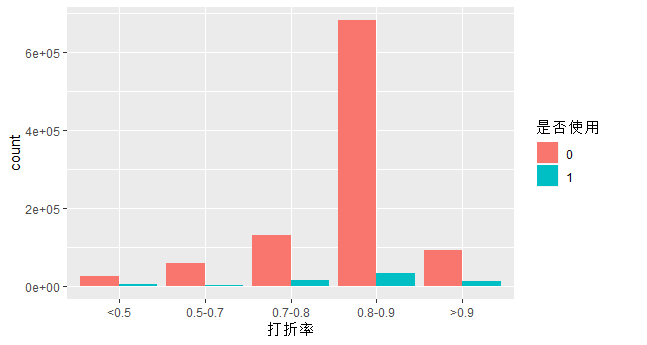
> prop.table(table(train\_1$use,rate\_cut),2)#计算使用率

rate\_cut

<0.5 0.5-0.7 0.7-0.8 0.8-0.9 >0.9

0 0.89185403 0.97681023 0.89841124 0.95382769 0.88458166

1 0.10814597 0.02318977 0.10158876 0.04617231 0.11541834



对比使用率和折扣的关系，看到不同折扣率的优惠券使用率有一定差别，但是并非折扣率越低使用率越高。

> IV(X=factor(train\_1$rate),Y=train\_1$use)#计算信息价值

[1] 0.4555317

attr(,"howgood")

[1] "Highly Predictive"

计算信息价值，显示为"Highly Predictive"。

（3）进一步来看使用门槛的影响

> #对比不同门槛的使用率

> ggplot(train\_1,aes(x=factor(train\_1$man),fill=factor(train\_1$use)))+

+ geom\_bar(position = "dodge")+

+ labs(x="使用门槛",fill="是否使用")

> prop.table(table(train\_1$use,train\_1$man),2)#计算使用率

0 5 10 20 30 50

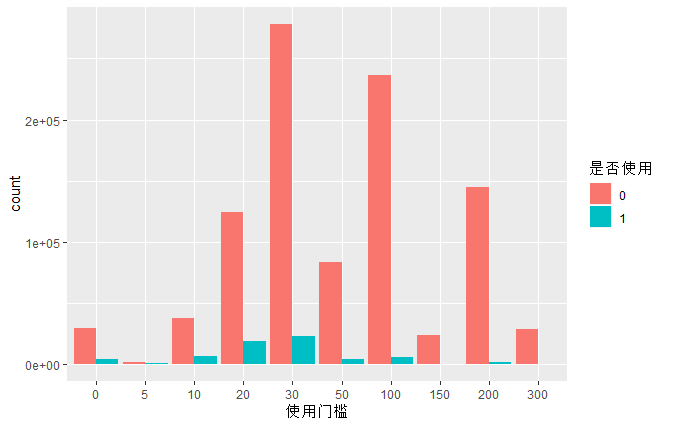
0 0.884226978 0.777117973 0.855073457 0.868716488 0.924502295 0.951748916

1 0.115773022 0.222882027 0.144926543 0.131283512 0.075497705 0.048251084

100 150 200 300

0 0.976259328 0.985545114 0.989740931 0.990329415

1 0.023740672 0.014454886 0.010259069 0.009670585



可以看到大部分优惠券的使用门槛越低，那么它的使用比例也高。

> IV(X=factor(train\_1$man),Y=train\_1$use)#计算信息价值

[1] 0.6988067

attr(,"howgood")

[1] "Highly Predictive"

进一步计算该特征的信息价值到了0.7，显示为"Highly Predictive"。

**2.3距离特征提取与分析**

> #对比不同距离对使用率的影响

> ggplot(train\_1,aes(x=train\_1$distance,fill=factor(train\_1$use)))+

+ geom\_bar(position = "dodge")+

+ labs(x="距离",fill="是否使用")

> prop.table(table(train\_1$use,train\_1$distance),2)#计算不同距离优惠券使用率

0 1 2 3 4 5

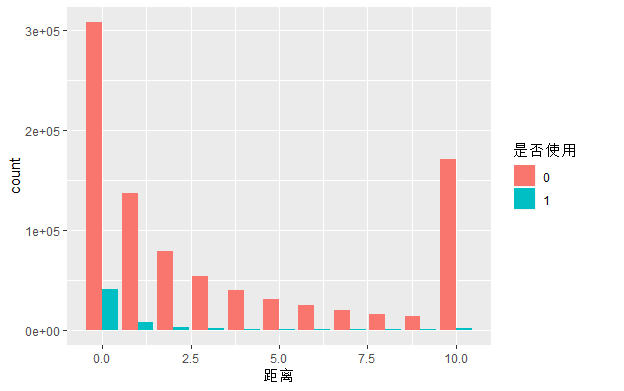
0 0.88430951 0.94987176 0.96471413 0.97291788 0.97432292 0.97807159

1 0.11569049 0.05012824 0.03528587 0.02708212 0.02567708 0.02192841

6 7 8 9 10

0 0.97996593 0.98343664 0.98055352 0.98101855 0.98792468

1 0.02003407 0.01656336 0.01944648 0.01898145 0.01207532



对比不同距离下的使用率，可以发现，距离越近的优惠券使用率越高。

> IV(X=factor(train\_1$distance),Y=train\_1$use)#计算信息价值

[1] 0.6853642

attr(,"howgood")

[1] "Highly Predictive"

进一步计算信息价值为0.68，属于"Highly Predictive"。

下面进行距离缺失值的插补。

> #计算距离缺失比例

> mean(is.na(train\_1$distance))

[1] 0.1006407

计算缺失值比例达到10%，缺失比例较大，采用回归插补法进行缺失值插补。因为距离和优惠券是否使用的相关度高，因此采用“距离”和“优惠券是否使用“进行线性回归的方式对距离缺失值进行回归插补。

> #距离缺失值回归插补

> a<-which(is.na(train\_1$distance))

> train\_1\_1<-train\_1[-a,]

> train\_1\_2<-train\_1[a,]

> model<-lm(distance~use,data = train\_1\_1)

> train\_1\_2$distance<-predict(model,train\_1\_2)

> train\_1<-rbind(train\_1\_1,train\_1\_2)

**2.4星期名特征的提取与分析**

> #增加领券的星期名

> train\_1$week\_r<-format(train\_1$date\_r,format = "%a")

> head(train\_1)

user shop coupon use distance date\_r date\_u type rate man week\_r

1 1439408 4663 11002 0 1 2016-05-28 <NA> 1 0.8666667 150 周六

2 1439408 2632 8591 0 0 2016-02-17 <NA> 1 0.9500000 20 周三

3 1439408 2632 1078 0 0 2016-03-19 <NA> 1 0.9500000 20 周六

4 1439408 2632 8591 0 0 2016-06-13 <NA> 1 0.9500000 20 周一

5 1439408 2632 8591 0 0 2016-05-16 2016-06-13 1 0.9500000 20 周一

6 1832624 3381 7610 0 0 2016-04-29 <NA> 1 0.9000000 200 周五

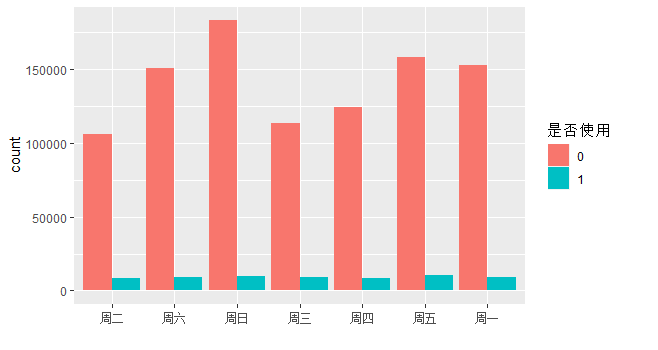
创建领券当天的星期名特征。

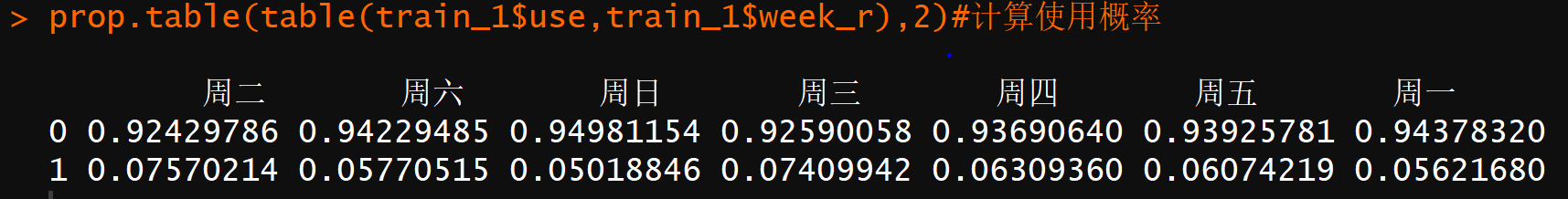
> #对比不同星期名的使用率

> ggplot(train\_1,aes(x=train\_1$week\_r,fill=factor(use)))+

+ geom\_bar(position = "dodge")+

+ labs(x="",fill="是否使用")





对比不同星期名对应的优惠券使用率，可以看到周末的使用率并没有比工作日更高。

> IV(X=factor(train\_1$week\_r),Y=train\_1$use)#计算信息价值

[1] 0.02094238

attr(,"howgood")

[1] "Not Predictive"

计算其信息价值，仅有0.02，为"Not Predictive"。综上可以看出，星期名对优惠券使用率影响不大。

**2.5优惠券日均领取次数的影响**

> #优惠券出现次数特征

> #计算优惠券的领取总次数

> coupon\_f<-as.data.frame(table(train\_1$coupon))

> coupon\_f$Var1<-as.integer(as.character(coupon\_f$Var1))

> #计算日均领取次数

> day\_1<-difftime(max(train\_1$date\_r),min(train\_1$date\_r),units="days")

> day\_1<-as.integer(day\_1)

> coupon\_f$Freq<-coupon\_f$Freq/day\_1

> #每张优惠券的日均领取次数赋值

> #构建赋值函数

> coupon\_cishu<-function(x)

+ {return(coupon\_f$Freq[which(coupon\_f$Var1==x)])}

> #赋值

> train\_1$coupon\_cishu<-sapply(train\_1$coupon,coupon\_cishu)

> head(train\_1)

user shop coupon use distance date\_r date\_u type rate man week\_r coupon\_cishu

1 1439408 4663 11002 0 1 2016-05-28 <NA> 1 0.8666667 150 周六 46.56626506

2 1439408 2632 8591 0 0 2016-02-17 <NA> 1 0.9500000 20 周三 0.18674699

3 1439408 2632 1078 0 0 2016-03-19 <NA> 1 0.9500000 20 周六 0.07228916

4 1439408 2632 8591 0 0 2016-06-13 <NA> 1 0.9500000 20 周一 0.18674699

5 1439408 2632 8591 0 0 2016-05-16 2016-06-13 1 0.9500000 20 周一 0.18674699

6 1832624 3381 7610 0 0 2016-04-29 <NA> 1 0.9000000 200 周五 281.50000000

创建每张优惠券的日领取次数特征，命名为coupon\_cishu，并将其添加到数据框中。

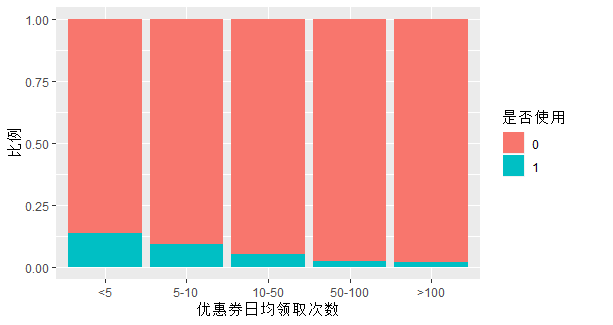
> #作图对比不同领取次数优惠券的使用率

> cut\_1<-cut(train\_1$coupon\_cishu,breaks = c(0,5,10,50,100,1000),labels =c("<5","5-10","10-50","50-100",">100"))

> ggplot(train\_1,aes(x=cut\_1,fill=factor(train\_1$use)))+

+ geom\_bar(position = "fill")+

+ labs(x="优惠券日均领取次数",y="比例",fill="是否使用")



作图对比不同领取次数的优惠券的使用率，可以看到，优惠券日均领取次数越多，其使用率反而越低。

> #计算信息价值

> IV(X=factor(train\_1$coupon\_cishu),Y=train\_1$use)

[1] 2.522624

attr(,"howgood")

[1] "Highly Predictive"

计算其信息价值，得到"Highly Predictive"的结论

**2.6商店名日均出现次数的影响**

> #商店名出现次数特征

> #计算不同商店的总出现次数

> shop\_f<-as.data.frame(table(train\_1$shop))

> shop\_f$Var1<-as.integer(as.character(shop\_f$Var1))

> #计算商店的日均出现次数

> shop\_f$Freq<-shop\_f$Freq/day\_1

> shop\_cishu<-function(x)

+ {return(shop\_f$Freq[which(shop\_f$Var1==x)])}

> #赋值

> train\_1$shop\_cishu<-sapply(train\_1$shop,shop\_cishu)

创建每张优惠券的日领取次数特征，命名为shop\_cishu，并将其添加到数据框中。

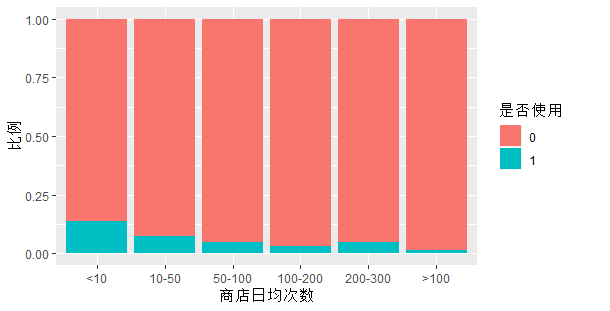
> #作图对比商店出现次数对使用率的影响

> cut\_2<-cut(train\_1$shop\_cishu,breaks = c(0,10,50,100,200,300,1000),labels =c("<10","10-50","50-100","100-200","200-300",">100"))

> ggplot(train\_1,aes(x=cut\_2,fill=factor(train\_1$use)))+

+ geom\_bar(position = "fill")+

+ labs(x="商店日均次数",y="比例",fill="是否使用")



总体上看，商店日均出现次数越高则使用率越低的分布规律。

> #计算信息价值

> IV(X=factor(train\_1$shop\_cishu),Y=train\_1$use)

[1] 1.74067

attr(,"howgood")

[1] "Highly Predictive"

计算其信息价值，为"Highly Predictive" 。

以上完成了特征工程的建立，共建立7个特征，按照各特征的信息价值排名为“日领取次数”>“商店名日出现次数”>“使用门槛”>“距离”>“折扣率”>“优惠券类型”>“星期名”。

**三、建立模型**

接下来建立机器学习模型，为了验证模型的性能，需要划分验证集进行模型验证，划分方式是按照领券日期，即训练集：20160101-20160515，验证集：20160516-20160615。

> #训练集1月1日-5月15日，测试集为5月16日-6月15日

> day\_1<-as.Date("2016-05-15")

> day\_2<-as.Date("2016-06-15")

> train\_data<-filter(train\_1,train\_1$date\_r<=day\_1)

> test\_data<-filter(train\_1,train\_1$date\_r>day\_1 & train\_1$date\_r<=day\_2)

构建模型，采用逻辑回归，并做预测。

> #逻辑回归

> model\_log<-glm(use~type+week\_r+rate+man+distance+coupon\_cishu+shop\_cishu,family=binomial(),data=train\_data)

> #预测

> a<-test\_data$use

> test\_data<-test\_data[,-4]

> predict<-predict(model\_log,type = "response",newdata = test\_data)

作出ROC曲线，并计算AUC为0.709

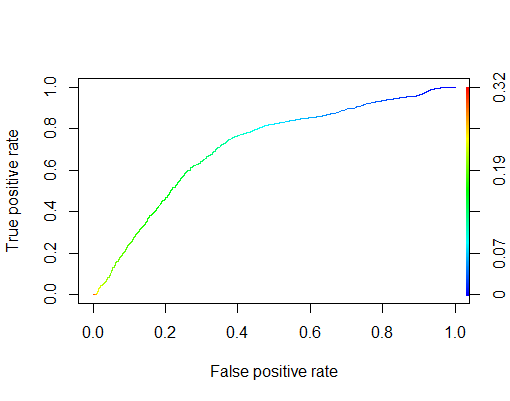
> #计算AUC

> pred<-prediction(predict,a)

> perf<-performance(pred,"tpr","fpr")

> plot(perf,colorize=T)

> performance(pred, "auc")



去掉信息价值较低的week\_r重新预测，auc略微增加到0.715，确认采用“日领取次数”，“商店名日出现次数”，“使用门槛”，“距离”，“折扣率”，“优惠券类型”这六个特征进行预测。

**四、预测**

构建最终测试集特征。

> #测试集特征构建

> #打折类型，打折率，使用门槛

> test$type<-parSapply(cl,test$discount,discounttype)

> test$rate<-parSapply(cl,test$discount,discountrate)

> test$man<-parSapply(cl,test$discount,man)

> #优惠券日均领取数量

> coupon\_f\_1<-as.data.frame(table(test$coupon))

> coupon\_f\_1$Var1<-as.integer(as.character(coupon\_f\_1$Var1))

> #计算日均领取次数

> coupon\_f\_1$Freq<-coupon\_f\_1$Freq/31

> #每张优惠券的日均领取次数赋值

> coupon\_cishu<-function(x)

+ {return(coupon\_f\_1$Freq[which(coupon\_f\_1$Var1==x)])}

> test$coupon\_cishu<-sapply(test$coupon,coupon\_cishu)

>

> #商店名出现次数

> shop\_f\_1<-as.data.frame(table(test$shop))

> shop\_f\_1$Var1<-as.integer(as.character(shop\_f\_1$Var1))

> #商店名日均出现次数赋值

> shop\_f\_1$Freq<-shop\_f\_1$Freq/31

> shop\_cishu<-function(x)

+ {return(shop\_f\_1$Freq[which(shop\_f\_1$Var1==x)])}

> test$shop\_cishu<-sapply(test$shop,shop\_cishu)

>

> #距离插补

> test[is.na(test$distance),]$distance<-mean(test$distance)

训练最终模型并预测

> #训练最终模型并预测

> model\_log<-glm(use~type+rate+man+distance+coupon\_cishu+shop\_cishu,family=binomial(),data=train\_1)

> predict<-predict(model\_log,type = "response",newdata = test)

导出预测结果。

> #导出预测结果

> submission<-select(test,user,coupon,date\_r)

> submission<-cbind(submission,predict)

> write.csv(submission, file = 'd:/data\_exercise/o2o/predict\_Solution.csv',row.names = F)

上传后查看分数及排名。