DATA ANALYTICS

QUESTION

You are Anne Actuary, a data scientist based in the country of Newland, working for Capland Express (also known as Capex), an international brand, famous for its charge cards, accepted all around the world.

It is 6pm on Friday 16th September 2016, and you are trying to finish up work for the week and go for a Friday night drink. Just as you start to shut down your computer, the phone rings. You answer, and hear the voice of Mark Ting, the Chief Marketing Officer. Mark is always excited and full of new ideas for more sales. This time Mark is excited about the upcoming release of the prestigious Capex Platinum card, an exclusive charge card with no monthly limit, available only to people with high incomes. Despite the fact that Capland Express is the market leader in most countries, due to legislative constraints that have recently been withdrawn, Capex has never been able to issue cards in Newland. This will be Capex's first product to ever be made available in Newland. Capex will start a huge marketing campaign on 31st October, with TV advertisements and whole page advertisements in luxury magazines. Mark has been frustrated that the card acceptance rules have not been finalised, and he has been making daily phone calls to you and your colleagues, urging you to finish your projects as soon as possible. This phone call is another of his reminders to hurry. You note that the project deadline is 29th September, with 9 working days remaining, and tell Mark some good news: the Newland Bureau of Statistics has just sent you the data you needed – sample data collected from Newland census and income tax data, and anonymised.

Mark has demanded that new Capex Platinum cards be issued within 1 minute of receiving the online application. Steve Vincent, the Senior Vice-President for Credit Risk, has said that this is not possible because:

1. Some customers don't have a high enough income to be accepted for a Platinum card
2. The overseas experience is that some people will be dishonest about their income
3. It takes 3 processing days to confirm a person's income

© 2016 Actuaries Institute

Since Mark and Steve couldn't agree, Catherine Everton-Overton, the CEO of Capland Express, had to mediate this dispute. Catherine tells them that Steve can set an acceptance rule that only requires income to be confirmed if there is a high chance that the person is overstating their income, and that Steve must not cause unnecessary delays in card acceptance. And that's how you got your project assignment – to recommend to Mark and Steve how to decide which card applicants are likely to be overstating their income. Yesterday Catherine noted that you were using external data, and she extended your scope to include recommendations on how to improve your model once Capex has begun collecting its own data.

Newland is a modern developed country with a population of approximately 50 million people. Its currency has a 1.012 exchange rate to the Australian dollar. It has similar data privacy and anti-discrimination laws to Australia.

Your report is due on Thursday 29th September 2016.

END OF QUESTION

Appendix A: Data Dictionary

Data Source: Newland Bureau of Statistics

Notes:

- Newland Bureau of Statistics created a custom dataset, linking census data and income tax data
- Due to privacy concerns, the Newland Bureau of Statistics is only willing to provide a small sample of data, with column choices carefully filtered to maintain anonymity
- The Newland Bureau of Statistics used a stratified sampling technique of the population of Newland, filtered to be in the age ranges defined by Capex
- The small / large split in the data is aligned with Capex minimum income requirements for a Platinum card

Column Name	Description
age	age of the person
workclass	employer classification
education	description of highest education status
educationnum	ranking of highest education status - 1 is lowest education, 16 is highest education status
maritalstatus	marital status
occupation	occupation classification
relationship	family status
race	ethnicity classification
sex	Gender
capitalgain	profits from investment sources, excluding wages / salary
capitalloss	losses from investment sources, excluding wages / salary
hoursperweek	number of hours worked per week
nativecountry	country of birth
income	whether the annual income is small or large

© 2016 Actuaries Institute

Appendix A: Data Dictionary

Note that you may choose whatever software you wish to do the data analysis. This script is not the technical solution, and is provided to give a starting point for those who are new to data analytics.

```
if (!require("pacman")) { install.packages("pacman"); library("pacman") }
pacman::p_load(data.table, party, rpart, rpart.plot, randomForest, gbm, glmnet, caret)

# read the case study data
# TO DO: use the path to your data file on your computer
training = read.csv('https://s3-ap-southeast-1.amazonaws.com/capcourse/CAP2016S2dataAnalytics.csv', stringsAsFactors = TRUE)

# are any of the target values NA? This would be a problem!
sum(is.na(training$income))

# check for missing values in the data
# random forest and regularized GLM will not work with missing values
data2 = training
col.has.na <- apply(data2, 2, function(x){any(is.na(x))})
if (sum(col.has.na) > 0)
{
  cat("Some columns have missing values\n")
  cat(names(data2)[col.has.na])
  cat("\n")
  # missing value flags
  for (colName in names(data2)[col.has.na])
  {
    cClass = class(data2[, colName])
    if (cClass == "factor") data2[, colName] = as.character(data2[, colName])
    if (cClass == "factor" || cClass == "character") data2[is.na(data2[, colName]), colName] = "_MISSING_VALUE_"
    if (cClass == "factor") data2[, colName] = as.factor(data2[, colName])
    if (cClass == "integer" || cClass == "numeric")
    {
      mu = mean(data2[, colName], na.rm = TRUE)
      newCol = paste0(colName, "_MISSING_FLAG")
      data2[, newCol] = ifelse(is.na(data2[, colName]), 1, 0)
      # imputed value
      data2[is.na(data2[, colName]), colName] = mu
    }
  }
}
rm(col.has.na)
rm(colName)
rm(mu)
rm(newCol)
rm(cClass)
gc()
```

```
# check for collinearity
# HINT: this could flag problems with your GLMs and regularized GLMs
mm = model.matrix(income ~ ., data = data2)
comboInfo = findLinearCombos(mm)
if (length(comboInfo$linearCombos) > 0)
{
  cat("Collinearity Found\n")
  cat("Linear Combinations Are:\n")
  colNames = colnames(mm)
  for (combo in comboInfo$linearCombos)
  {
    cat(paste0("  ", paste(colNames[combo], collapse = "  ")))
    cat("\n")
  }
  if (length(comboInfo$remove) > 0)
  {
    cat("Suggested Column Removals:\n")
    for (rem in comboInfo$remove)
      cat(paste0("   ", colNames[rem], "\n"))
  }
}
rm(mm)
rm(comboInfo)
rm(colNames)
rm(combo)
rm(rem)
gc()

# make a version of the data where the target is numeric {0, 1}
data3 = training
data3$isHigh = ifelse(data3$income == "large", 1, 0)
data3$income = NULL
for (i in seq_len(ncol(data3)))
{
  if (class(data3[, i]) == "character")
    data3[, i] = as.factor(data3[, i])
}
for (i in seq_len(ncol(data3)))
{
  if (sum(is.na(data3[,i])) > 0) print(names(data3)[i])
}
for (i in seq_len(ncol(data2)))
{
  if (sum(is.na(data2[,i])) > 0) print(names(data2)[i])
}
# and a version where the target is numeric, and the missing values are imputed
data4 = data2
data4$isHigh = ifelse(data4$income == "large", 1, 0)
data4$income = NULL
for (i in seq_len(ncol(data4)))
{
```

```
  if (class(data4[, i]) == "character")


data3[, i] = as.factor(data4[, i])
}
for (i in seq_len(ncol(data4)))
{
  if (sum(is.na(data4[,i])) > 0) print(names(data4)[i])
}
for (i in seq_len(ncol(data2)))
{
  if (sum(is.na(data4[,i])) > 0) print(names(data4)[i])
}

# build a decision tree on proportions
model.rpart = rpart(income ~ ., data = training)
prp(model.rpart, type = 2)
# build a second decision tree on probabilities
model.rpart.2 = rpart(isHigh ~ ., data = data3)
prp(model.rpart.2, type = 2)

# build a different type of decision tree
model.ctree = ctree(isHigh ~ ., data = data3, controls = ctree_control(minsplit = 1000,
minbucket = 750))
plot(model.ctree, type = "simple")

# reclaim RAM for the next model type
rm(model.rpart)
rm(model.ctree)
gc()

# build a random forest
set.seed(1234)
model.rf = randomForest(income ~ ., data = data2, proximity = FALSE, nodesize = 10,
maxnodes = 5)
importances = importance(model.rf)
print(importances)
varImpPlot(model.rf)

# reclaim RAM for the next model type
rm(model.rf)
rm(importances)
gc()

# build a gradient boosted trees model - not compulsory in the syllabus but an excellent
substitute for random forest
# HINT: for this case study the GBM gives some very useful insights
set.seed(1234)
model.gbm = gbm(isHigh ~ ., data = data3, cv.folds = 5, n.cores = 2, n.trees = 5000)
# check performance using 5-fold cross-validation
best.iter <- gbm.perf(model.gbm, method="cv")
print(best.iter)
# plot variable importance
```

```
summary(model.gbm, n.trees=best.iter) # based on the estimated best number of trees

# create marginal plots
# plot variable X1,X2,X3 after "best" iterations

par(mfrow=c(1,3))
plot(model.gbm, 5, best.iter)
plot(model.gbm, 8, best.iter)
plot(model.gbm, 11, best.iter)
par(mfrow=c(1,1))
# contour plot of variables 5 and 8
plot(model.gbm, c(5, 8), best.iter)
# contour plot of variables 5 and 11
plot(model.gbm, c(5, 11), best.iter)

# reclaim RAM for the next model type
rm(model.gbm)
rm(best.iter)
gc()

# build a regularized logistic GLM
#
# glmnet does not support a formula interface
# so need to manually create the design matrix
# GLMs don't cope with missing values, so use the imputed values
i = which(names(data4) == "isHigh", arr.ind = TRUE)
# create the model matrix with on-hot-encoding
x = model.matrix(isHigh ~ ., data4)
# fit the model - please be patient as this may take a long time
set.seed(1234)
model.rglm = cv.glmnet(x = x, y = as.matrix(data4$isHigh), family = "binomial", nfolds = 5,
maxit = 1000)
plot(model.rglm)
#
model.rglm2 = glmnet(x = x, y = as.matrix(data4$isHigh), family = "binomial", maxit = 1000)
plot(model.rglm2, label = TRUE)
#
plot(model.rglm2, xvar = "dev", label = TRUE)
# list the linear coefficients
coef(model.rglm, s = "lambda.min")

# reclaim RAM for the next model type
rm(i)
rm(x)
rm(model.rglm)
rm(model.rglm2)
gc()
```

```
# build a traditional glm
# GLMs don't cope with missing values, so use the imputed values
# HINT: you should see warnings about "fitted probabilities numerically 0 or 1 occurred" -
these should alert you to a possible problem
model.glm = glm(isHigh ~ ., data = data4, family = "binomial")
# show the coefficients and their significance
summary(model.glm)
# show the Akaike Information Criterion
print(model.glm$aic)
# show the Bayesian Information Criterion
BIC(model.glm)
# do we have any NA coefficients? this warns us of separation or collinearity
model.glm$coefficients[is.na(model.glm$coefficients)]
# diagnistics on residuals
plot(model.glm)

# reclaim RAM for the next model type
rm(model.glm)
gc()
```