

Supplementay

Skills and Vacancy Analysis with Data Mining Techniques (R Code)

Data Exploration

```
ij2014 <- read.csv("ij2014.csv", header=TRUE)
dim(ij2014)
class(ij2014)
colnames(ij2014)
str(ij2014)

ij2014$JobTitle <- as.character(ij2014$JobTitle)
ij2014$JobDescription <- as.character(ij2014$JobDescription)

percent <- function (x) {
  per <- cbind(table(x), round(prop.table(table(x))*100,2))
  colnames(per) <- c('Count', 'Percentage')
  per
}

nlevels(ij2014$CategoryNames)
percent(ij2014$CategoryNames)
ij2014$CategoryNames<-as.character(ij2014$CategoryNames)
ij2014$CategoryNames<- gsub(",Public Sector","", ij2014$CategoryNames)
ij2014$CategoryNames<-as.factor(ij2014$CategoryNames)
nlevels(ij2014$CategoryNames)
percent(ij2014$CategoryNames)

it2014 <- ij2014[ij2014$CategoryNames == "IT", ]
it2014 <- it2014[,c("JobId", "JobTitle", "JobDescription")]
dim(it2014)
class(it2014)
colnames(it2014)
str(it2014)

it2014$JobDescription <-gsub("<(.|\\n)*?>", " ", it2014$JobDescription)
it2014$JobDescription <-gsub("&nbsp;|&lsquo;|&rsquo;|&amp;|&ldquo;|&rdquo;", " ",
it2014$JobDescription)
it2014$JobDescription <- gsub("[[:punct:]]", " ", it2014$JobDescription)
it2014$JobTitle <- gsub("[[:punct:]]", " ", it2014$JobTitle)

library(tm)

myStopwords <- c("job", "work", "dublin", "location", "employment", "call", "contact", "will",
"required", "year", "years", "candidate", "e g")

corpusT <- Corpus(VectorSource(it2014$JobTitle))
corpusD <- Corpus(VectorSource(it2014$JobDescription))

corpusT <- tm_map(corpusT, tolower)
corpusT <- tm_map(corpusT, removewords, stopwords('english'))
corpusT <- tm_map(corpusT, removeNumbers)
corpusT <- tm_map(corpusT, removewords, myStopwords)
corpusT <- tm_map(corpusT, stripwhitespace)

corpusD <- tm_map(corpusD, tolower)
corpusD <- tm_map(corpusD, removewords, stopwords('english'))
corpusD <- tm_map(corpusD, removeNumbers)
corpusD <- tm_map(corpusD, removewords, myStopwords)
corpusD <- tm_map(corpusD, stripwhitespace)

corpusT <- tm_map(corpusT, PlainTextDocument)
matrixT <- TermDocumentMatrix(corpusT)
matrixD <- removeSparseTerms(matrixD, 0.99)

corpusD <- tm_map(corpusD, PlainTextDocument)
matrixD <- TermDocumentMatrix(corpusD)
matrixD <- removeSparseTerms(matrixD, 0.99)

MatrixTfreq <- rowSums(as.matrix(matrixT))
MatrixTfreq<-sort(MatrixTfreq, decreasing = TRUE)
MatrixTtop30<- MatrixTfreq [1:30]

MatrixDfreq <- rowSums(as.matrix(matrixD))
MatrixDfreq<-sort(MatrixDfreq, decreasing = TRUE)
MatrixDtop30<- MatrixDfreq [1:30]

library(lattice)
library(ggplot2)

tiff("JobTitle.tiff", width = 7, height = 5, units = "in", res = 300)
barplot(MatrixTtop30, space=0, col='cadetblue1', las=1, cex.lab=0.3, cex.axis=0.5, cex.names=0.6,
horiz=TRUE)
dev.off()

tiff("JobDescription.tiff", width = 7, height = 5, units = "in", res = 300)
```

```

barplot(MatrixDtop30, space=0, col='forestgreen', las=2, cex.lab=0.6, cex.axis=0.5, cex.names=0.6,
horiz=TRUE)
dev.off()

findAssocs(matrixT, 'manager', 0.10)
findAssocs(matrixT, 'support', 0.15)
findAssocs(matrixT, 'senior', 0.07)
findAssocs(matrixD, 'skills', 0.25)
findAssocs(matrixD, 'experience', 0.25)
findAssocs(matrixD, 'knowledge', 0.20)
findAssocs(matrixD, 'ability', 0.25)

save.image()

```

Data Mining

```

Support <- subset(it2014, grep1("Consultant|Support", it2014$JobTitle))
Support$Label <- "Support"

Engineer <- subset(it2014, grep1("Engineer", it2014$JobTitle))
Engineer$Label <- "Engineer"

Analyst <- subset(it2014, grep1("Analyst", it2014$JobTitle))
Analyst$Label <- "Analyst"

Lead <- subset(it2014, grep1("Manager|Lead", it2014$JobTitle))
Lead$Label <- "Lead"

Administrator <- subset(it2014, grep1("Administrator|Admin", it2014$JobTitle))
Administrator$Label <- "Administrator"

Tester <- subset(it2014, grep1("Test|Tester|Quality|QA", it2014$JobTitle))
Tester$Label <- "Tester"

Developer <- subset(it2014, grep1("Developer|Architect", it2014$JobTitle))
Developer$Label <- "Developer"

it2014_lab <- rbind(Support, Engineer, Analyst, Lead, Administrator, Tester, Developer)
dup.idx <- it2014_lab[duplicated(it2014_lab$JobId),1]
it2014_lab <- it2014_lab[!(it2014_lab$JobId %in% dup.idx), ]
it2014_nolab <- it2014 [!(it2014$JobId %in% it2014_lab$JobId), ]
it2014_nolab<-it2014_nolab[!duplicated(it2014_nolab$JobId), ]
intersect(it2014_lab$JobId, it2014_nolab$JobId)

tiff("labels.tiff", width = 7, height = 5, units = "in", res = 300)
barplot(table(it2014_lab$Label), space=0.2, main='IT Labels', col=c("tan1", "violetred4", "black",
"seagreen4", "seashell3", "red3", "wheat4"),
las=2, cex.lab=0.6, cex.names=0.6, horiz=TRUE, xlim=c(0,1400))
dev.off()

transform <- function(x){
  c <- Corpus(VectorSource(x[["JobDescription"]]))
  c <- tm_map(c, tolower)
  c <- tm_map(c, stemDocument, language = 'english')
  c <- tm_map(c, removeWords, stopwords('english'))
  c <- tm_map(c, removeWords, myStopwords)
  c <- tm_map(c, removeNumbers)
  c <- tm_map(c, stripwhitespace)
  c <- tm_map(c, PlainTextDocument)
  matrixc <- TermDocumentMatrix(c)
  matrixc <- removeSparseTerms(matrixc, 0.999)
  matrixcT <- t(matrixc)
  matrixcdf <- as.data.frame(inspect(matrixcT))
  rownames(matrixcdf) <- as.vector(x[["JobId"]])
  return (as.data.frame(matrixcdf))
}

idlab <- it2014_lab[,c(idlab <- it2014_lab[,c("JobId", "JobTitle", "Label")])
it2014_lab <- it2014_lab[,c("JobId", "JobDescription")]
itMatrix_lab <-transform(it2014_lab)

idlab_nolab <- it2014_nolab[,c("JobId", "JobTitle")]
it2014_nolab <- it2014_nolab[,c("JobId", "JobDescription")]
itMatrix_nolab <-transform(it2014_nolab)

library(plyr)

AllData_list <- list(itMatrix_lab, itMatrix_nolab)
AllData <- do.call(rbind.fill, AllData_list)
AllData[is.na(AllData)] <- 0
ModelData <- AllData[1:4755,]
UnseenData <- AllData[4756:6983,]
train.idx <- sample(nrow(ModelData), ceiling(nrow(ModelData) * 0.8))
test.idx <- (1:nrow(ModelData)) [-train.idx]

library(class)

k=c(1,2,3,4,5,7,9,11,13,15)
p=rep(0,10)
summary=cbind(k,p)

```

```

colnames(summary)=c("k", "Percent Classified Correctly")

for(i in 1:10){
  result <- knn(ModelData[train.idx, ], ModelData[test.idx, ], idlab[train.idx,"Label"], k=i)
  summary[i,2]=sum(diag(table(result,idlab[test.idx, "Label"]))/length(test.idx) * 100)
}

summary

tiff("plotknn.tiff", width = 7, height = 5, units = "in", res = 300)
plot(summary, pch=16, cex.axis=0.8, col="red")
lines(summary, lty=3, col="red")
dev.off()

knn1 <- knn(ModelData[train.idx, ], ModelData[test.idx, ], idlab[train.idx,"Label"], k=1)
knn1_table <- table( "Prediction" = knn1, Job = idlab[test.idx,"Label"])
knn1_table

test.true <- idlab[test.idx,]
knn1<- as.character(knn1)
predictions <- cbind(test.true, knn1 )
percent(knn1_pred$Label)
percent(knn1_pred$knn1)
counts <- cbind(table(predictions$Label), table(predictions$knn1))

tiff("knn1_class.tiff", width = 7, height = 5, units = "in", res = 300)
barplot(t(counts), col=c("tan1", "tan1", "violetred4", "violetred4", "black", "black", "seagreen4",
"seagreen4", "seashell3", "seashell3", "red3", "red3", "wheat4", "wheat4"), beside=TRUE, cex.names=0.6, cex.axis=0.7, ylim=c(0,350))
grid(NA, 15, lty = 1, lwd = 0.5, col = "gray")
dev.off()

```

New Data Predictions

```

knn1_UnseenData <- knn(ModelData[train.idx, ], UnseenData, idlab[train.idx,"Label"], k=1)
UnseenData_predictions <- cbind(idlab_nolab, it2014_nolab$JobDescription, knn1_UnseenData)
colnames(UnseenData_predictions)[3] <- "JobDescription"
UnseenData_predictions$JobDescription <- as.character(UnseenData_predictions$JobDescription)
UnseenData_predictions[1:20, c(2,4) ]

tiff("knn1_unseen.tiff", width = 7, height = 5, units = "in", res = 300)
barplot(table(knn1_UnseenData), col=c( "tan1", "violetred4", "black","seagreen4","seashell3", "red3",
"wheat4"), beside=TRUE, ylim=c(0,800),
cex.axis=0.8, cex.names=0.8)
grid(NA, 15, lty = 1, lwd = 0.5, col = "gray")
dev.off()

Specialist <- subset(UnseenData_predictions, grepl("Specialist", UnseenData_predictions$JobTitle))

tiff("knn1_specialist.tiff", width = 7, height = 5, units = "in", res = 300)
barplot(table( Specialist$knn1_UnseenData), col=c( "tan1", "violetred4", "black", "seagreen4",
"seashell3", "red3", "wheat4"), beside=TRUE,
ylim=c(0,70),cex.axis=0.8, cex.names=0.8)
grid(NA, 15, lty = 1, lwd = 0.5, col = "gray")
dev.off()

save.image()

```

Skills Visualizations

```

AdministratorP <- subset(UnseenData_predictions, grepl("Administrator", knn1_UnseenData))
AnalystP <- subset(UnseenData_predictions, grepl("Analyst", knn1_UnseenData))
DeveloperP <- subset(UnseenData_predictions, grepl("Developer", knn1_UnseenData))
EngineerP <- subset(UnseenData_predictions, grepl("Engineer", knn1_UnseenData))
LeadP <- subset(UnseenData_predictions, grepl("Lead", knn1_UnseenData))
SupportP <- subset(UnseenData_predictions, grepl("Support", knn1_UnseenData))
TesterP <- subset(UnseenData_predictions, grepl("Test", knn1_UnseenData))

library(wordcloud)
library(Rweka)

BigramTokenizer <- function(x) NGramTokenizer(x, weka_control(min = 2, max = 2))
transformBi <- function(x) {
  c <- Corpus(VectorSource(x[["JobDescription"]]))
  c <- tm_map(c, tolower)
  c <- tm_map(c, removeWords, stopwords('english'))
  c <- tm_map(c, removeWords, myStopwords)
  c <- tm_map(c, removeNumbers)
  c <- tm_map(c, stripwhitespace)
  c <- tm_map(c, PlainTextDocument)
  matrixc <- TermDocumentMatrix(c, control = list(tokenize = BigramTokenizer))
  matrixc <- removeSparseTerms(matrixc, 0.999)
}

AdMBi<-transformBi(AdministratorP)
AnMBi<-transformBi(AnalystP)
DevMBi<-transformBi(DeveloperP)
EnMBi<-transformBi(EngineerP)

```

```

LeMBi<-transformBi(LeadP)
SuMBi<-transformBi(SupportP)
TeMBi<-transformBi(TesterP)

AdMFBi <- rowSums(as.matrix(AdMBi))
AdMFBi<-sort(AdMFBi, decreasing = TRUE)
AdMFBiTop<- AdMFBi [1:20]
tiff ("AdMFBiTop.tiff", width=12, height=8, units="in", res=300)
wordcloud(words=names(AdMFBiTop), freq=AdMFBiTop, random.order=T, col=" tan1")
dev.off()

AnMFBi <- rowSums(as.matrix(AnMBi))
AnMFBi<-sort(AnMFBi, decreasing = TRUE)
AnMFBiTop<- AnMFBi [1:20]
tiff ("AnMFBiTop. tiff ", width=12, height=8, units="in", res=300)
wordcloud(words=names(AnMFBiTop), freq=AnMFBiTop, random.order=T, col=" violetred4")
dev.off()

DevMFBi <- rowSums(as.matrix(DevMBi))
DevMFBi<-sort(DevMFBi, decreasing = TRUE)
DevMFBiTop<- DevMFBi [1:20]
tiff ("DevMFBiTop.tiff ", width=12, height=8, units="in", res=300)
wordcloud(words=names(DevMFBiTop), freq=DevMFBiTop, random.order=T, col=" black")
dev.off()

EnMFBi <- rowSums(as.matrix(EnMBi))
EnMFBi<-sort(EnMFBi, decreasing = TRUE)
EnMFBiTop<- EnMFBi [1:20]
tiff ("EnMFBiTop.tiff ", width=12, height=8, units="in", res=300)
wordcloud(words=names(EnMFBiTop), freq= EnMFBiTop, random.order=T, col=" seagreen4")
dev.off()

LeMFBi <- rowSums(as.matrix(LeMBi))
LeMFBi<-sort(LeMFBi, decreasing = TRUE)
LeMFBiTop<- LeMFBi [1:20]
tiff ("LeMFBiTop.tiff ", width=12, height=8, units="in", res=300)
wordcloud(words=names(LeMFBiTop), freq= LeMFBiTop, random.order=T, col=" seashell4")
dev.off()

SuMFBi <- rowSums(as.matrix(SuMBi))
SuMFBi<-sort(SuMFBi, decreasing = TRUE)
SuMFBiTop<- SuMFBi [1:20]
tiff ("SuMFBiTop.tiff ", width=12, height=8, units="in", res=300)
wordcloud(words=names(SuMFBiTop), freq= SuMFBiTop, random.order=T, col=" red3")
dev.off()

TeMFBi <- rowSums(as.matrix(TeMBi))
TeMFBi<-sort(TeMFBi, decreasing = TRUE)
TeMFBiTop<- TeMFBi [1:20]
tiff ("TeMFBiTop.tiff ", width=12, height=8, units="in", res=300)
wordcloud(words=names(TeMFBiTop), freq= TeMFBiTop, random.order=T, col=" wheat4")
dev.off()

```