Analytical Occupation Classfication on O*NET Database

Akshay Kulkarni

25 March 2019

Introduction

The ONET Database: A Primary Source of Occupational Information

The ONET database has a wide variety of worker and job oriented data categories. The ONET Content Model provides the framework that identifies and organizes this important information about work. The O*NET-SOC Occupation Taxonomy covers work performed in the U.S. economy and defines the set of occupations for which data is collected.

The ONet is now the primary source of occupational information. It is sponsored by ETA through a grant to the North Carolina Department of Commerce. Thus, it is a rich database with a sizable ampunt of information of very high quality.

This is script deals with the task of classifying a given occupation from the the ONET database by implementing a model to estimate a probability of an occupation being *Analytical* in nature. After spending some time looking around on the ONET site, Relevant tables that contain possible promising features were identified and data was aqucisitoned, cleaned and transformed to create a feature matrix on which modelling was done to predict an occupation being analytical.

This RMD script contains:

- Feature Analysis and manipulation.
- Training the model on manually pre-labled data
- Prediction

Checking and Loading required libraries

```
# Initialization

packages <- c("tidyverse","readxl","scales","mice","randomForest","MASS","caret","klaR","xlsx","modelr"

checkPackage <- function(package_vec){                  # defining a custom function for checking packag
                            for (p in package_vec){
                                if(p %in% rownames(installed.packages()) == FALSE){
                                    cat(paste(p,"Package is not found/installed on this machine,
                                    install.packages(p,dependencies = TRUE) # Installing with dependanc
                                } else {
                                cat(paste("[",p,"]","is present. \n"))
                              }
                             }
                           }
```

```
checkPackage(packages) # running check
```

```
## [ tidyverse ] is present.
## [ readxl ] is present.
## [ scales ] is present.
## [ mice ] is present.
## [ randomForest ] is present.
## [ MASS ] is present.
## [ caret ] is present.
## [ klaR ] is present.
## [ xlsx ] is present.
## [ modelr ] is present.
## [ kernlab ] is present.
```

```
# Load packages

library('MASS')
library('readxl') # Reading data
library('scales') # visualization
library('dplyr') # data manipulation (already loaded with Tidyverse)
library('mice') # needed for possible imputation
library('randomForest') # RF classification algorithm
library('tidyverse') # Data manipulation + Visualization
library('klaR') # for partimat function for plotting discriminant analysis plots
library('xlsx') # writing to xlsx format
library('modelr') # for partitioning function
library('caret')
library('kernlab') #Gaussian Process Classification Function
```

Data collation and Feature Selection:

Before any feature selection can begin setting a base definition about what constitutes as an ANALYTICAL and what aspects might influence the nature of an occupation will be good.

Solutions can be reached by clear-cut, methodical approaches or more creative and lateral angles, depending on the objective. Both ways of solving a problem require analytical skills. Analytical skills might sound technical, but we use these skills in everyday work when detecting patterns, brainstorming, observing, interpreting data, integrating new information, theorizing, and making decisions based on multiple factors and options available.

therefore features that could capture the analytical nature of an occupation would have descriptions strongly correlating to aspects in :

- Communication.
- Creativity.
- Critical Thinking & Pattern analysis.
- Information Processing.
- Analytics
- Research.

etc.

Therefore after systematically observing all tables int the ONET Content model and correlating features by using the Content Model Reference, Various variables from multiple tables such as Ablities, Skills, Work Activities , Interests were identified and further reduced to avoid repeated or correlating features.

Thus the final feature list was identified as below :

SOC Code	Selected Feature	Description
1.A.1.b.2	Originality	The ability to come up with unusual or clever ideas about a given topic or situation, or to develop creative ways to solve a problem.
1.A.1.b.3	Problem Sensitivity	The ability to tell when something is wrong or is likely to go wrong. It does not involve solving the problem, only recognizing there is a problem.
1.A.1.b.5	Inductive Reasoning	The ability to combine pieces of information to form general rules or conclusions (includes finding a relationship among seemingly unrelated events).
1.C.7.a	Innovation	Job requires creativity and alternative thinking to develop new ideas for and answers to work-related problems.
1.C.7.b	Analytical Thinking	Job requires analyzing information and using logic to address work-related issues and problems.
2.A.2.a	Critical Thinking	Using logic and reasoning to identify the strengths and weaknesses of alternative solutions, conclusions or approaches to problems.
2.A.2.b	Active Learning	Understanding the implications of new information for both current and future problem-solving and decision-making.
2.B.2.i	Complex Problem Solving	Identifying complex problems and reviewing related information to develop and evaluate options and implement solutions.
4.A.2.a.1	Judging the Qualities of Things, Services, or People	Assessing the value, importance, or quality of things or people.
4.A.2.a.4	Analyzing Data or Information	Identifying the underlying principles, reasons, or facts of information by breaking down information or data into separate parts.
4.A.2.b.1	Making Decisions and Solving Problems	Analyzing information and evaluating results to choose the best solution and solve problems.
4.A.2.b.3	Updating and Using Relevant Knowledge	Keeping up-to-date technically and applying new knowledge to your job.
4.A.2.b.4	Developing Objectives and Strategies	Establishing long-range objectives and specifying the strategies and actions to achieve them.

12 variables have been identified that describe analytical attributes. These features do a pretty good job of encompassing the analytical nature of an occupation title from various aspects or facets mentioned in our base definition/assumption.

```
# Loading data (make sure the tables re in the same fold as the rmd)

script_folder <-  getwd() # Retrieving the path from where the rmd is being accessed (To eliminate loca


occupation_rawdata <- read_xlsx(paste(getwd(),"/Occupation Data.xlsx",sep = "")) # loading Occupation t

skills_rawdata <-  read_xlsx(paste(getwd(),"/Skills.xlsx",sep = ""))  # loading Skills table

scale_ref <-  read_xlsx(paste(getwd(),"/Scales Reference.xlsx",sep = "")) # loading Scales ref table
```

```
blsdata <-  read_xlsx(paste(getwd(),"/national_M2017_dl.xlsx",sep = "")) # national BLS data for genera

abilities_rawdata <-  read_xlsx(paste(getwd(),"/Abilities.xlsx",sep = "")) # loading abilities table

workactivities_rawdata <- read_xlsx(paste(getwd(),"/Work Activities.xlsx",sep = "")) # loading work act

interests_rawdata <- read_xlsx(paste(getwd(),"/Interests.xlsx",sep = "")) # loading interests table

workstyles_rawdata <- read_xlsx(paste(getwd(),"/Work Styles.xlsx",sep = ""))

CMR <-  read_xlsx(paste(getwd(),"/Content Model Reference.xlsx",sep = "")) # Content model Reference de

blsdatamay18 <- blsdata %>% mutate(OCC_CODE = paste(OCC_CODE,".00",sep="")) # BLS data for possible EDA
```

```
str(skills_rawdata)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    67690 obs. of  15 variables:
##  $ O*NET-SOC Code    : chr  "11-1011.00" "11-1011.00" "11-1011.00" "11-1011.00" ...
##  $ Title             : chr  "Chief Executives" "Chief Executives" "Chief Executives" "Chief Executive
##  $ Element ID        : chr  "2.A.1.a" "2.A.1.a" "2.A.1.b" "2.A.1.b" ...
##  $ Element Name      : chr  "Reading Comprehension" "Reading Comprehension" "Active Listening" "Acti
##  $ Scale ID          : chr  "IM" "LV" "IM" "LV" ...
##  $ Scale Name        : chr  "Importance" "Level" "Importance" "Level" ...
##  $ Data Value        : num  4.12 4.75 4.12 4.88 4 4.38 4.38 4.88 3.25 3.62 ...
##  $ N                 : num  8 8 8 8 8 8 8 8 8 ...
##  $ Standard Error    : num  0.13 0.16 0.13 0.23 0 0.18 0.18 0.13 0.16 0.26 ...
##  $ Lower CI Bound    : num  3.88 4.43 3.88 4.43 4 4.02 4.02 4.63 2.93 3.11 ...
##  $ Upper CI Bound    : num  4.37 5.07 4.37 5.32 4 4.73 4.73 5.12 3.57 4.14 ...
##  $ Recommend Suppress: chr  "N" "N" "N" "N" ...
##  $ Not Relevant      : chr  NA "N" NA "N" ...
##  $ Date              : chr  "07/2014" "07/2014" "07/2014" "07/2014" ...
##  $ Domain Source     : chr  "Analyst" "Analyst" "Analyst" "Analyst" ...
```

```
library(tidyverse)

skills <- skills_rawdata %>% select(`O*NET-SOC Code`,Title,`Element Name`,`Scale ID`,`Data Value`)%>%
                             filter(`Scale ID` == "LV")%>%
                               spread(key = `Element Name`, value = `Data Value`)


abilities <- abilities_rawdata %>% select(`O*NET-SOC Code`,Title,`Element Name`,`Scale ID`,`Data Value`
                                  spread(key = `Element Name`, value = `Data Value`)

activities <- workactivities_rawdata %>% select(`O*NET-SOC Code`,Title,`Element Name`,`Scale ID`,`Data
                            spread(key = `Element Name`, value = `Data Value`)

interests <- interests_rawdata %>% select(`O*NET-SOC Code`,Title,`Element Name`,`Scale ID`,`Data Value`
                            spread(key = `Element Name`, value = `Data Value`)
```

```r
styles <- workstyles_rawdata %>% select(`O*NET-SOC Code`,Title,`Element Name`,`Scale ID`,`Data Value`)%
                            spread(key = `Element Name`, value = `Data Value`)
```

```r
merged_features_data <- skills %>% left_join(abilities,by =c("O*NET-SOC Code","Title")) %>% left_join(a
```

```r
#-----------------------------------------------------------------------------------------------------
```

```r
features <-  c("O*NET-SOC Code","Title",
               "Originality",
               "Problem Sensitivity",
               "Inductive Reasoning",
               "Innovation",
               "Analytical Thinking",
               "Critical Thinking",
               "Active Learning",
               "Complex Problem Solving",
               "Judging the Qualities of Things, Services, or People",
               "Making Decisions and Solving Problems",
               "Updating and Using Relevant Knowledge",
               "Developing Objectives and Strategies")  # Defining a vector of chosen features
```

```r
main_features <- merged_features_data %>% dplyr::select(one_of(features)) # picking cols/features of in
```

```r
head(main_features,10)
```

```
## # A tibble: 10 x 14
##    `O*NET-SOC Code` Title Originality `Problem Sensit~ `Inductive Reas~ Innovation `Analytical Thi~
##    <chr>            <chr>       <dbl>            <dbl>            <dbl>      <dbl>            <dbl>
##  1 11-1011.00       Chie~        4.25             5                5         4.27             4.45
##  2 11-1011.03       Chie~        4                4.25             4.25      4.38             4.31
##  3 11-1021.00       Gene~        3.38             3.88             3.38      3.65             4.03
##  4 11-2011.00       Adve~        3.88             4                3.88      3.99             3.88
##  5 11-2021.00       Mark~        4                3.88             3.88      4.15             4
##  6 11-2022.00       Sale~        3.88             3.88             4         4.13             4.14
##  7 11-2031.00       Publ~        4                4                3.75      3.68             3.79
##  8 11-3011.00       Admi~        3.12             3.75             3.38      3.85             4.04
##  9 11-3021.00       Comp~        3.5              4                4         3.7              4.25
## 10 11-3031.01       Trea~        3.38             4.75             4.12      3.57             4.63
## # ... with 7 more variables: `Critical Thinking` <dbl>, `Active Learning` <dbl>, `Complex Problem
## #   Solving` <dbl>, `Judging the Qualities of Things, Services, or People` <dbl>, `Making
## #   Decisions and Solving Problems` <dbl>, `Updating and Using Relevant Knowledge` <dbl>,
## #   `Developing Objectives and Strategies` <dbl>
```

```
write_csv(main_features,paste(getwd(),"/TrainingFeatures.csv",sep=""))
```

Thus we have obtained a clean feature matrix with each row forming a feature vector $\hat{x} \in \mathbb{R}^{12}$ for $n = 967$ occupation titles

Generating Lables for Training

Our feature matrix is missing the binary response variable $y \in 0, 1$ which denotes whether a particular job is `Analytical`

Therefore the response variable was manually generated by hand-labeling. The subjective assessment was done with randomly sampling occupations and comparing them to a psuedo-rubric consisting the base assumption/definition and the attributes that are indicative of the analytical nature of an occupation.

I surmised that the procedure would best carried out together with a cohort of MSDS students (a group of 5 Friends) at Northeastern University in order to mitigate the unavoidable subjective bias in the data which was natrually bound to arise. (hand-labelled 167 occupations, assigning 1 if analytical, and 0 if not). The resulting DF contained the response variable `ANALYTICAL` $\in 0, 1$ otherwise `NA` for unlablled points.

Loading Labled data

```
Labled_original_data <-  read_csv(paste(getwd(),"/LabledTrainingFeatures.csv",sep = ""), na = "NA")
```

```
## Parsed with column specification:
## cols(
##   `O*NET-SOC Code` = col_character(),
##   Title = col_character(),
##   ANALYTICAL = col_double(),
##   Originality = col_double(),
##   `Problem Sensitivity` = col_double(),
##   `Inductive Reasoning` = col_double(),
##   Innovation = col_double(),
##   `Analytical Thinking` = col_double(),
##   `Critical Thinking` = col_double(),
##   `Active Learning` = col_double(),
##   `Complex Problem Solving` = col_double(),
##   `Judging the Qualities of Things, Services, or People` = col_double(),
##   `Making Decisions and Solving Problems` = col_double(),
##   `Updating and Using Relevant Knowledge` = col_double(),
##   `Developing Objectives and Strategies` = col_double()
## )
```

```
model_data <- Labled_original_data
names(model_data)<-str_replace_all(names(model_data), c(" " = ".", "," = ""))# renaming cols and removi

str(model_data)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 967 obs. of  15 variables:
##  $ O*NET-SOC.Code                                : chr  "11-1011.00" "11-1011.03" "11-1021.00" "
```

```
##  $ Title                                                : chr  "Chief Executives" "Chief Sustainability
##  $ ANALYTICAL                                           : num  1 1 1 NA NA NA NA NA 1 NA ...
##  $ Originality                                          : num  4.25 4 3.38 3.88 4 3.88 4 3.12 3.5 3.38
##  $ Problem.Sensitivity                                  : num  5 4.25 3.88 4 3.88 3.88 4 3.75 4 4.75 ..
##  $ Inductive.Reasoning                                  : num  5 4.25 3.38 3.88 3.88 4 3.75 3.38 4 4.12
##  $ Innovation                                           : num  4.27 4.38 3.65 3.99 4.15 4.13 3.68 3.85 3
##  $ Analytical.Thinking                                  : num  4.45 4.31 4.03 3.88 4 4.14 3.79 4.04 4.25
##  $ Critical.Thinking                                    : num  4.75 4.12 4 4.12 4.25 4 4 3.88 4 4.5 ...
##  $ Active.Learning                                      : num  4.75 3.75 3.62 4.12 4.12 3.88 3.5 3.38 3
##  $ Complex.Problem.Solving                              : num  5 4.25 3.75 3.88 3.88 3.88 3.88 3.12 3.88
##  $ Judging.the.Qualities.of.Things.Services.or.People: num  5.39 4.27 4.43 3.66 3.53 3.96 4.09 3.94 4
##  $ Making.Decisions.and.Solving.Problems                : num  6.18 5.4 4.48 4.18 4.67 5.26 4.6 4.67 5.1
##  $ Updating.and.Using.Relevant.Knowledge                : num  4.92 5.54 4.53 4.49 4.52 4.87 4.93 4.99 5
##  $ Developing.Objectives.and.Strategies                 : num  5.68 5.12 3.77 3.9 4.17 4.78 4.8 3.34 4.0
##  - attr(*, "spec")=
##   .. cols(
##   ..   `O*NET-SOC Code` = col_character(),
##   ..   Title = col_character(),
##   ..   ANALYTICAL = col_double(),
##   ..   Originality = col_double(),
##   ..   `Problem Sensitivity` = col_double(),
##   ..   `Inductive Reasoning` = col_double(),
##   ..   Innovation = col_double(),
##   ..   `Analytical Thinking` = col_double(),
##   ..   `Critical Thinking` = col_double(),
##   ..   `Active Learning` = col_double(),
##   ..   `Complex Problem Solving` = col_double(),
##   ..   `Judging the Qualities of Things, Services, or People` = col_double(),
##   ..   `Making Decisions and Solving Problems` = col_double(),
##   ..   `Updating and Using Relevant Knowledge` = col_double(),
##   ..   `Developing Objectives and Strategies` = col_double()
##   .. )
```

Modelling.

Randomforest.

As a non-parametric modelling technique `randomForest` theoretically should and does perform well on classifcation problems and since its easy to implement its a great way to build a preliminary model to test initial accuracy and gather more information on importance of the selected features.

```
# RandomForest Analysis

rfdata <- model_data %>% drop_na() # doriiping NA rows basically gives us the part of the data that we

set.seed(55)  # for reproducibitlity

partitions <- resample_partition(rfdata,c(train = 0.7,test=0.3)) # partitioning labeled data for traini

train_df <- as.tibble(partitions$train) %>% dplyr::select(-`O*NET-SOC.Code`,-`Title`) # removing title
```

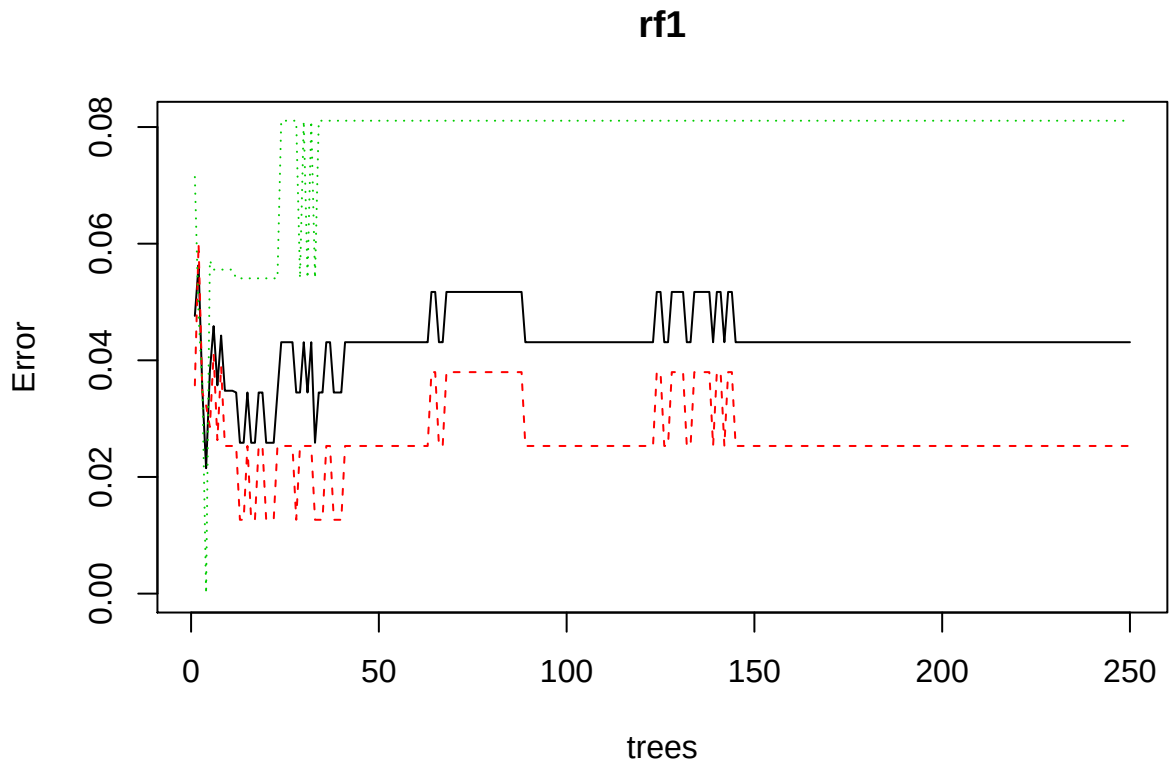
```
## Warning: `as.tibble()` is deprecated, use `as_tibble()` (but mind the new semantics).
```

```
## This warning is displayed once per session.
```

```
test_df <- as.tibble(partitions$test)  # coercing to tibble if train/test need to be viewed.

rf1 <- randomForest(as.factor(ANALYTICAL) ~ . , data = train_df, mtry=12 , ntree = 250, keep.forest=TRU
rf1
```

```
##
## Call:
##  randomForest(formula = as.factor(ANALYTICAL) ~ ., data = train_df,      mtry = 12, ntree = 250, keep
##                Type of random forest: classification
##                      Number of trees: 250
## No. of variables tried at each split: 12
##
##         OOB estimate of  error rate: 4.31%
## Confusion matrix:
##    0  1 class.error
## 0 77  2  0.02531646
## 1  3 34  0.08108108
```

```
plot(rf1)
```




```
importance(rf1)
```

```
##                                                MeanDecreaseGini
```

```
## Originality                                              0.09589807
## Problem.Sensitivity                                      0.85456702
## Inductive.Reasoning                                      5.64588497
## Innovation                                               0.21223543
## Analytical.Thinking                                      1.99927346
## Critical.Thinking                                        1.59110887
## Active.Learning                                          2.12232543
## Complex.Problem.Solving                                 34.44832829
## Judging.the.Qualities.of.Things.Services.or.People       0.81493694
## Making.Decisions.and.Solving.Problems                    1.23524875
## Updating.and.Using.Relevant.Knowledge                    0.83692841
## Developing.Objectives.and.Strategies                     0.08333333
```

```
# Calculating predictions from train/test of labled data for randomForest

model_test_pred <- test_df %>% mutate("Prediction" = format(predict(rf1,test_df),scientific = FALSE))

model_test_pred_classwise <- test_df %>% mutate("Pred Prob for (0)" = format(predict(rf1,test_df,type =
                                  "Pred Prob for (1)" = format(predict(rf1,test_df,type = "prob")[,2],



model_test_pred <- subset(model_test_pred, select=c(`O*NET-SOC.Code`, `Title`,`ANALYTICAL`,`Prediction`

model_test_pred_classwise <- subset(model_test_pred_classwise, select=c(`O*NET-SOC.Code`, `Title`,`ANAL




# Displaying Confusion Matrix

Cmatrix1 <- confusionMatrix(as.factor(model_test_pred$Prediction),as.factor(model_test_pred$ANALYTICAL)
Cmatrix1
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 27  4
##          1  2 18
##
##                Accuracy : 0.8824
##                  95% CI : (0.7613, 0.9556)
##     No Information Rate : 0.5686
##     P-Value [Acc > NIR] : 1.29e-06
##
##                   Kappa : 0.7575
##
##  Mcnemar's Test P-Value : 0.6831
##
##             Sensitivity : 0.8182
##             Specificity : 0.9310
##          Pos Pred Value : 0.9000
##          Neg Pred Value : 0.8710
```

```
##              Prevalence : 0.4314
##          Detection Rate : 0.3529
##    Detection Prevalence : 0.3922
##       Balanced Accuracy : 0.8746
##
##        'Positive' Class : 1
##
```

```
cat("The Accuracy is",Cmatrix1$overall["Accuracy"])
```

```
## The Accuracy is 0.8823529
```

```
# Predicting values for original partially-labled data

model_full_pred <- model_data %>% mutate("Prediction" = format(predict(rf1,model_data),scientific = FALS

model_full_pred_classwise <- model_data %>% mutate("Pred Prob for (0)" = format(predict(rf1,model_data,
                                    "Pred Prob for (1)" = format(predict(rf1,model_data,type = "prob")[,1



model_full_pred <- subset(model_full_pred, select=c(`O*NET-SOC.Code`, `Title`,`ANALYTICAL`,`Prediction`

model_full_pred_classwise <- subset(model_full_pred_classwise, select=c(`O*NET-SOC.Code`, `Title`,`ANAL


write.xlsx(model_full_pred,paste(getwd(),"/rfPredResult.xlsx",sep=""))
write.xlsx(model_full_pred_classwise,paste(getwd(),"/rfPredResult(classwise).xlsx",sep=""))
```

The model obtained an 88% accuracy against the test data and the importance function indicates that the `Complex Problem Solving` feature/attribute is the most influential variable in the model indicated by its highest meandecreaseGINI where the scale is irrelevant: only the relative values matter.

(note: the results with predictions and class probs are saved in the cwd)

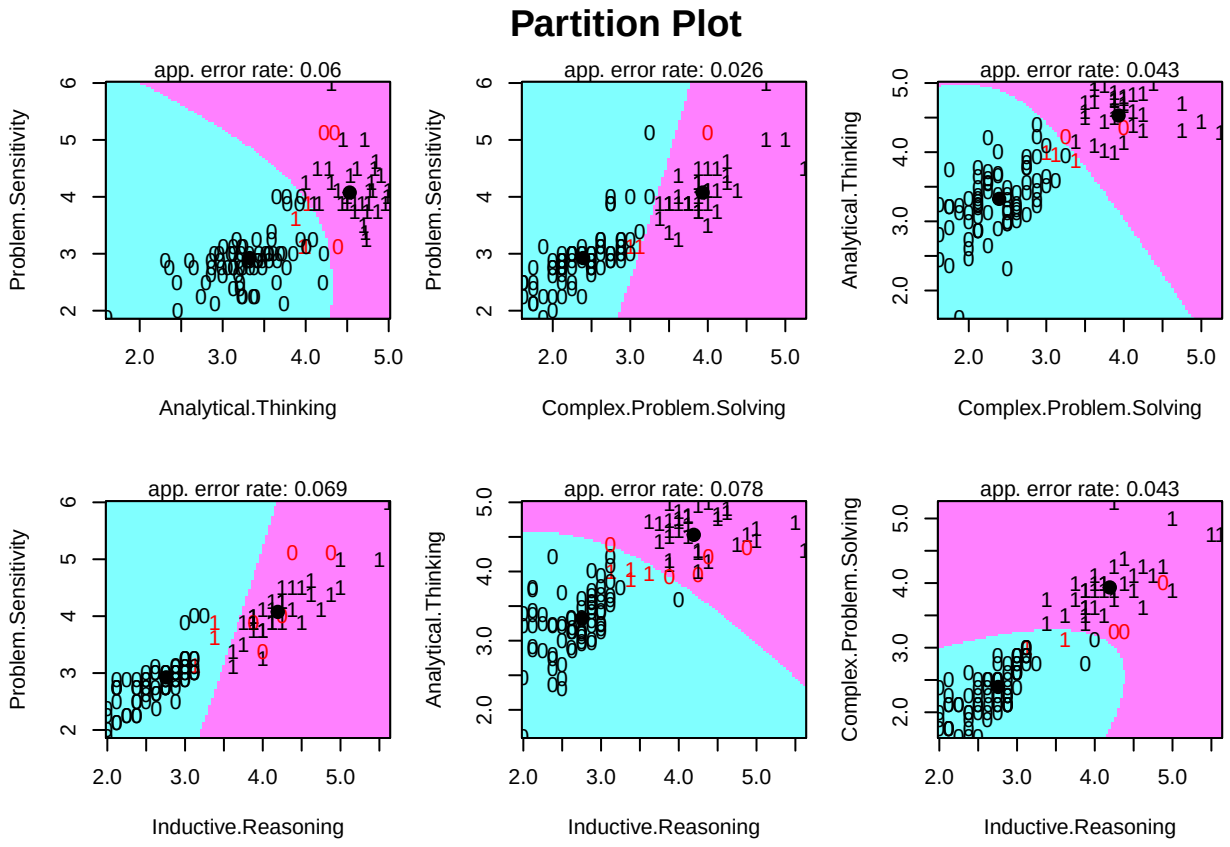
Quadratic Discriminant Analysis.

```
# Quadratic Discriminant Analysis.

qdadata <- model_data %>% drop_na()

qda.model <- qda(ANALYTICAL ~ . , train_df)
qda.model
```

```
## Call:
## qda(ANALYTICAL ~ ., data = train_df)
##
## Prior probabilities of groups:
##         0         1
## 0.6810345 0.3189655
##
```

```
## Group means:
##   Originality Problem.Sensitivity Inductive.Reasoning Innovation Analytical.Thinking
## 0    2.072658            2.924810            2.756709   3.309747             3.324937
## 1    3.605135            4.075135            4.192973   3.850000             4.530270
##   Critical.Thinking Active.Learning Complex.Problem.Solving
## 0          2.859367        2.328608                2.391899
## 1          4.189459        3.963243                3.932703
##   Judging.the.Qualities.of.Things.Services.or.People Making.Decisions.and.Solving.Problems
## 0                                           2.912025                                3.402405
## 1                                           4.171081                                5.215676
##   Updating.and.Using.Relevant.Knowledge Developing.Objectives.and.Strategies
## 0                              3.402532                             2.151646
## 1                              5.447568                             4.057297
```

```
partimat(as.factor(ANALYTICAL) ~ Problem.Sensitivity+Analytical.Thinking+Complex.Problem.Solving+Inducti
```




```
qda.test <- predict(qda.model,test_df)
test_df$qda <- qda.test$class


Cmatrix2 <- confusionMatrix(as.factor(test_df$qda),as.factor(test_df$ANALYTICAL), dnn = c("Prediction",
Cmatrix2
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction  0  1
##          0 28  3
##          1  1 19
##
##                Accuracy : 0.9216
##                  95% CI : (0.8112, 0.9782)
##     No Information Rate : 0.5686
##     P-Value [Acc > NIR] : 2.904e-08
##
##                   Kappa : 0.8384
##
##  Mcnemar's Test P-Value : 0.6171
##
##             Sensitivity : 0.9655
##             Specificity : 0.8636
##          Pos Pred Value : 0.9032
##          Neg Pred Value : 0.9500
##              Prevalence : 0.5686
##          Detection Rate : 0.5490
##    Detection Prevalence : 0.6078
##       Balanced Accuracy : 0.9146
##
##        'Positive' Class : 0
##
```

```
cat("The Accuracy is",Cmatrix2$overall["Accuracy"])
```

```
## The Accuracy is 0.9215686
```

```
# Predicting values for original partially-labled data

model_full_pred2 <- model_data %>% mutate("Prediction" = (predict(qda.model,model_data))$class) # predi
```

```
## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning Inf
```

```
model_full_pred2 <- subset(model_full_pred2, select=c(`O*NET-SOC.Code`, `Title`,`ANALYTICAL`,`Predictio


write.xlsx(model_full_pred2,paste(getwd(),"/QDAPredResult.xlsx",sep=""))
```

The model obtained an 92% accuracy which is an improvement over our rF model. The `partimat()` fucntion provides a multiple figure array which shows the classification of observations based on classification methods (lda, qda, rpart, naiveBayes, rda, sknn and svmlight) for every combination of two variables. Moreover, the classification boundaries are displayed and the apparent error rates are given in each title.

Gaussian Process Classifier.

Now we get to the main part of this excercise and implement a `Gaussian Process Classifier`.

```r
# Gaussian Process Classifier.

gpc_model <- gausspr(as.factor(ANALYTICAL) ~ ., data = train_df, type= 'classification', kernel="anovad
          kpar="automatic", var=1, variance.model = FALSE, tol=0.0005,
          cross=0, fit=TRUE, na.action = na.omit)
```

```
##  Setting default kernel parameters
```

```r
(gpc_model)
```

```
## Gaussian Processes object of class "gausspr"
## Problem type: classification
##
## Anova RBF kernel function.
##  Hyperparameter : sigma =  1 degree =  1
##
## Number of training instances learned : 116
## Train error : 0.017241379
```

```r
gpc_data_test <- as.tibble(test_df)

gpc_data_test$Pred <- predict(gpc_model,gpc_data_test[,-3]) # need to remove the response variable from

# class probabilities

Cmatrix3 <- confusionMatrix(gpc_data_test$Pred,as.factor(gpc_data_test$ANALYTICAL), dnn = c("Prediction
Cmatrix3
```

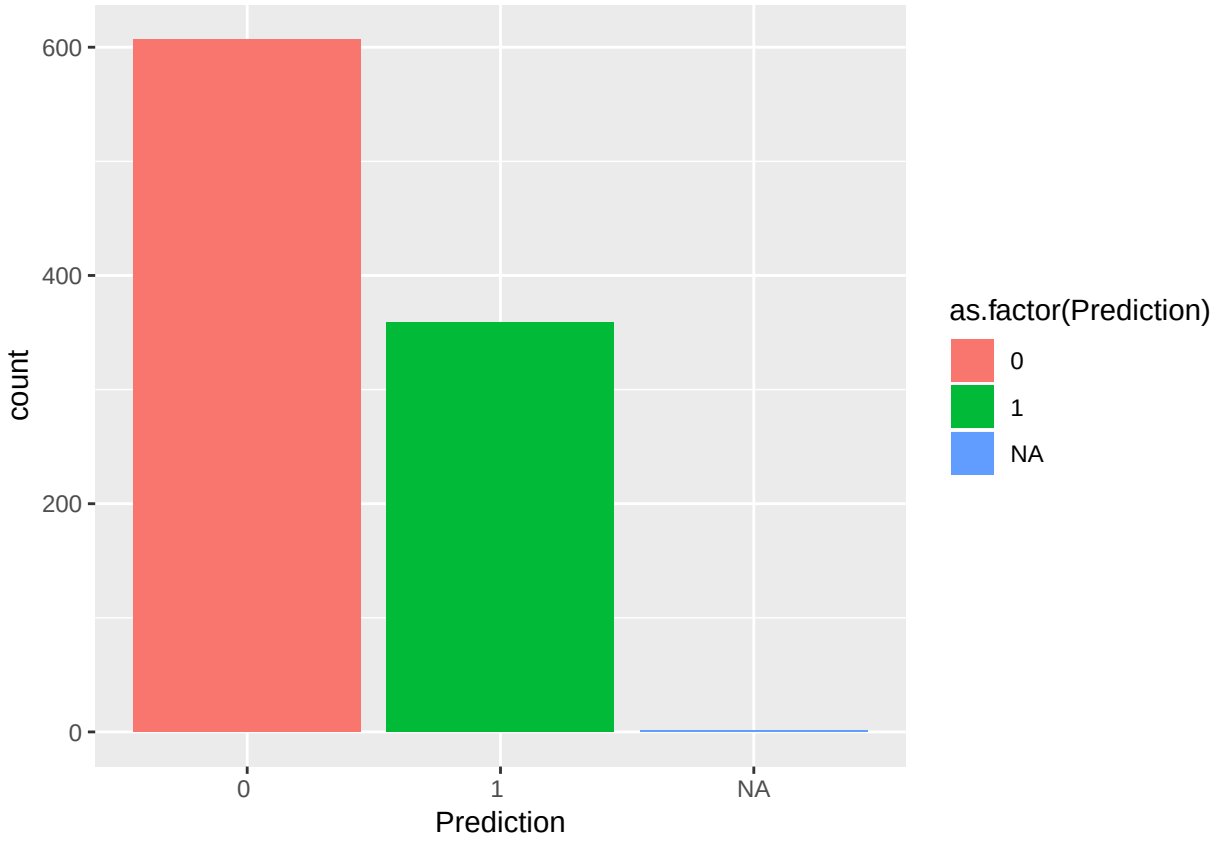
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 27  0
##          1  2 22
##
##                Accuracy : 0.9608
##                  95% CI : (0.8654, 0.9952)
##     No Information Rate : 0.5686
##     P-Value [Acc > NIR] : 2.425e-10
##
##                   Kappa : 0.9209
##
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 0.9310
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9167
##              Prevalence : 0.5686
##          Detection Rate : 0.5294
##    Detection Prevalence : 0.5294
##       Balanced Accuracy : 0.9655
```

```
##
##        'Positive' Class : 0
##
```

```
cat("The Accuracy is",Cmatrix3$overall["Accuracy"])
```

```
## The Accuracy is 0.9607843
```

```
# Predicting values for original partially-labled data

model_full_pred3 <- model_data %>% mutate("Prediction" = format(predict(rf1,model_data),scientific = FA

model_full_pred_classwise <- model_data %>% mutate("Pred Prob for (0)" = format(predict(rf1,model_data,
                                   "Pred Prob for (1)" = format(predict(rf1,model_data,type = "prob")[,1



model_full_pred3 <- subset(model_full_pred3, select=c(`O*NET-SOC.Code`, `Title`,`ANALYTICAL`,`Predictio



write.xlsx(model_full_pred3,paste(getwd(),"/GPCPredResult.xlsx",sep=""))

ggplot(data = filter(model_full_pred3, !is.na(model_full_pred3$Prediction)))+
  geom_bar(aes(x=Prediction,fill = as.factor(Prediction)))
```



The gaussian classifier is far outperforming our other models with 96% accuracy while there should be more analysis done before we can concretely state anything but it does confirm Gaussian Process as a well known powerful and highly flexible classifier.

The Gaussian process (GP) directly captures the model uncertainty.e.g for regression GP directly gives you a distribution for the prediction value, rather than just one value as the prediction. This uncertainty is not directly captured in neural networks.

It :

Can learn the kernel parameters automatically from data, no matter how flexible we wish to make the kernel. Can learn the regularization parameter C without cross-validation. *Can incorporate interpretable noise models and priors over functions, and can sample from prior to get intuitions about the model assumptions.* We can combine automatic feature selection with learning using ARD.

Conclusions:

ONET as a database has proven be a very informative and resourceful for occupation classification. It contains rich and detailed feature information on occupations and serves as a great repository fo analyses.

Q2b. Evaluation of the results. Do they make sense? What occupations came up as analytical that surprised you? What occupations did you expect to be analytical and didn't come up? What other sources of data could be included to make the model more accurate?

Answer - Most of the occupations are in accordance with our base definition and are predicted resonably accurately,and the barlpot indicates there are more non-analytical jobs than analytical in the ONET database, and Surprisingly jobs such as "Brodcast News Analyst" & "Cytogenetic Technician" are marked non analytical while "Nurse Midwives" are analytical which is a strech thus there could be some correlation or features we could be missing.

Q2c. The assumptions you made and anything you would consider going back and changing in the model.

Answer - The models can be tuned further and maybe a different choice of kernel can be used for GPC but i would invest more time into feature analysis, selection and generation to improve the models.

```
# some data analysis

edadata <-  model_full_pred3

feature_level_ref <-  read_xlsx(paste(getwd(),"/Level Scale Anchors.xlsx",sep = ""))

feature_level_ref <-  feature_level_ref %>% filter(`Element Name` %in%(features))

edadata$Prediction <- as.integer(edadata$Prediction)
```

```
## Warning: NAs introduced by coercion
```

```
inaccurate <- edadata %>% filter(ANALYTICAL!= Prediction & !is.na(ANALYTICAL))
inaccurate
```

```
## # A tibble: 6 x 16
##   `O*NET-SOC.Code` Title ANALYTICAL Prediction Originality Problem.Sensiti~ Inductive.Reaso~
##   <chr>            <chr>      <dbl>      <int>       <dbl>            <dbl>            <dbl>
## 1 11-9031.00       Educ~          0          1         3.5              4                3.88
## 2 13-2011.01       Acco~          1          0         3                3.75             3.88
## 3 27-3021.00       Broa~          1          0         3.5              4                3.88
## 4 29-1161.00       Nurs~          0          1         3                5                4.12
## 5 29-2011.01       Cyto~          1          0         2.62             4.12             4.12
## 6 41-3031.02       Sale~          1          0         2.88             3.62             3.5
## # ... with 9 more variables: Innovation <dbl>, Analytical.Thinking <dbl>, Critical.Thinking <dbl>,
## #   Active.Learning <dbl>, Complex.Problem.Solving <dbl>,
## #   Judging.the.Qualities.of.Things.Services.or.People <dbl>,
## #   Making.Decisions.and.Solving.Problems <dbl>, Updating.and.Using.Relevant.Knowledge <dbl>,
## #   Developing.Objectives.and.Strategies <dbl>
```

```
x <- ggplot(data = filter(edadata, !is.na(model_full_pred3$Prediction)))+
  geom_bar(aes(x=Prediction,fill = as.factor(Prediction)))

library(plotly) # comment it out if needed
```

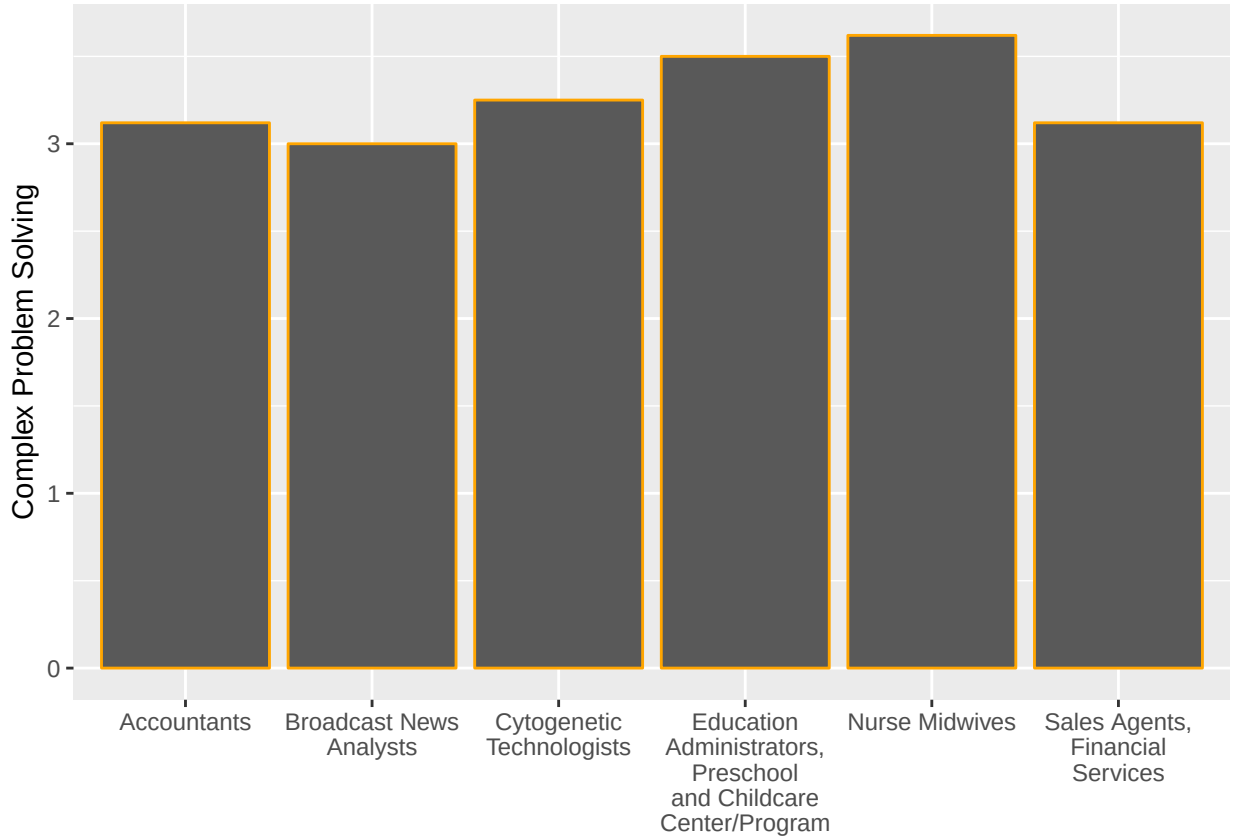
```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:MASS':
##
##     select

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```
inaccurate_pred <- inaccurate %>% inner_join(occupation_rawdata, by = c("O*NET-SOC.Code"="O*NET-SOC Code

select(inaccurate_pred,Title,Complex.Problem.Solving)
```

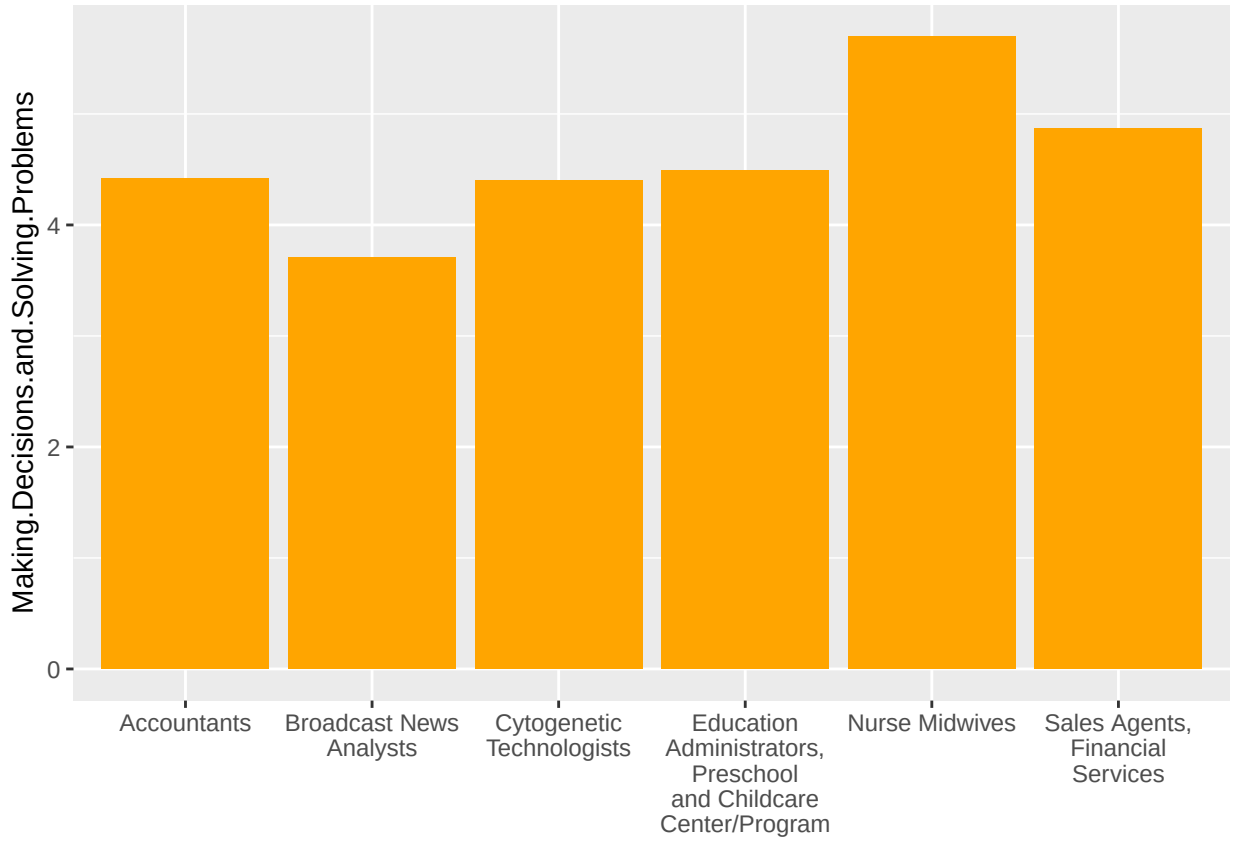
```
## # A tibble: 6 x 2
##   Title                                                                Complex.Problem.Solving
##   <chr>                                                                                  <dbl>
## 1 Education Administrators, Preschool and Childcare Center/Program                         3.5
## 2 Accountants                                                                              3.12
```

```
## 3 Broadcast News Analysts                                                3
## 4 Nurse Midwives                                                         3.62
## 5 Cytogenetic Technologists                                              3.25
## 6 Sales Agents, Financial Services                                       3.12
```

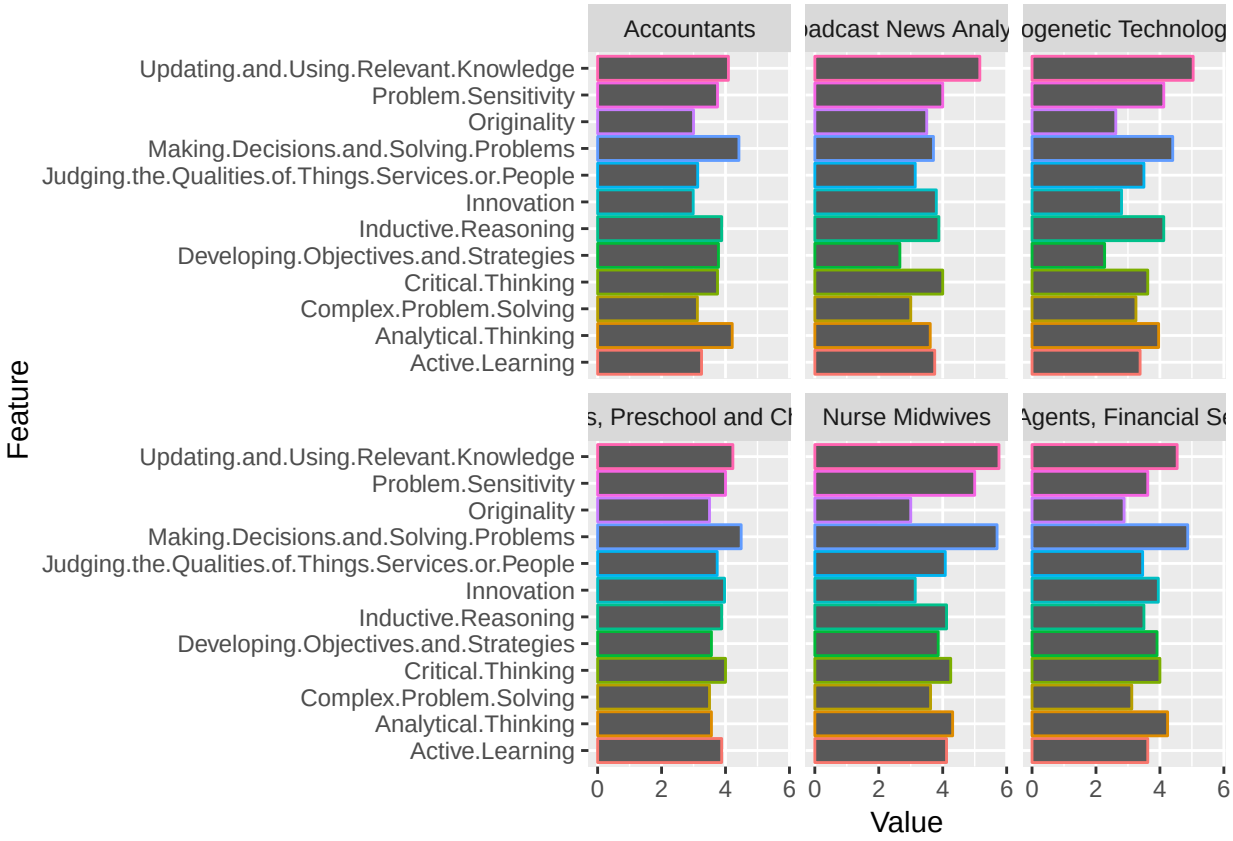
```
test <- inaccurate %>% gather("Feature", "Value", 5:16) # reshaping the df for plotting


ggplot(data= filter(test, Feature == "Complex.Problem.Solving"),aes(x=Title,y=Value))+
         geom_bar(stat = "identity",color="orange")+
        aes(stringr::str_wrap(Title, 15)) + xlab(NULL) +
        ylab("Complex Problem Solving")
```




```
ggplot(data= filter(test, Feature == "Making.Decisions.and.Solving.Problems"),aes(x=Title,y=Value))+
         geom_bar(stat = "identity",fill="orange")+
        aes(stringr::str_wrap(Title, 15)) + xlab(NULL) +
        ylab("Making.Decisions.and.Solving.Problems")
```




```
#plotting all feature values

ggplot(dplyr::group_by(test,Title), aes(y=Value, x=Feature ,color=Feature)) +
  geom_bar(stat="identity",show.legend = FALSE)+
  facet_wrap(.~Title)+
  coord_flip()
```


Accountants
adcast News Analy
ogenetic Technolog
s, Preschool and Cl
Nurse Midwives
Agents, Financial Se
Updating.and.Using.Relevant.Knowledge
Problem.Sensitivity
Originality
Making.Decisions.and.Solving.Problems
Judging.the.Qualities.of.Things.Services.or.People
Innovation
Inductive.Reasoning
Developing.Objectives.and.Strategies
Critical.Thinking
Complex.Problem.Solving
Analytical.Thinking
Active.Learning
Feature
0
2
4
6
Value