Data Analysis and Prediction of Questioner Segments

Kjersti Framnes

2023-08-24

Introduction

This dataset consists of respondents' answers to a series of questions, along with their demographic details such as age, gender, and county. The variables of interest for our analysis, columns `E` through `P`, represent responses to various questions and are primarily categorical in nature.

The primary goal of this project is to predict the `Segment` column, which categorizes respondents based on their responses. The secondary goal is to see if any of these models can exceed a precision of 70%. This because my company needs to see if they can achive the same segments based on 11 questions. They already have a model that requires 30 questions and there is a high cost to ask this many background questions when interviewing

To achieve this, we will:

1. Conduct thorough data exploration to understand the distribution and relationships among variables.
2. Use machine learning models, specifically Random Forest, k-Nearest Neighbors (kNN), and Naive Bayes, to predict the `Segment` column.
3. Evaluate the performance of each model to determine its effectiveness and see if they exceed 70%.

With this understanding, we will proceed to the methods and analysis section.

First we need to make sure we have all the required packages, libraries and to read the data. The data is located on my Github account.

```
if(!require(readxl)) install.packages("readxl", repos = "http://cran.us.r-project.org")
if(!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(randomForest)) install.packages("randomForest", repos = "http://cran.us.r-project.org")
if(!require(kknn)) install.packages("kknn", repos = "http://cran.us.r-project.org")
if(!require(e1071)) install.packages("e1071", repos = "http://cran.us.r-project.org")
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(formatR)) install.packages("formatR", repos = "http://cran.us.r-project.org")
if(!require(RColorBrewer)) install.packages("RColorBrewer", repos = "http://cran.us.r-project.org")
if(!require(tidyr)) install.packages("tidyr", repos = "http://cran.us.r-project.org")
if(!require(gt)) install.packages("gt", repos = "http://cran.us.r-project.org")

# Load required libraries
library(readxl)
library(dplyr)
```

```
library(ggplot2)
library(caret)
library(randomForest)
library(kknn)
library(e1071)
library(tidyverse)
library(formatR)
library(RColorBrewer)
library(tidyr)
library(gt)

knitr::opts_chunk$set(echo = TRUE,warning = FALSE,size="tiny", tidy = TRUE,message=FALSE)


# Required Libraries
library(readxl)

# Download Excel file from GitHub to a temporary local path
url <- "https://raw.githubusercontent.com/kjerstif/PH125.9x-Data-Science-Capstone/main/Capstone_data.xls
local_path <- tempfile(fileext = ".xlsx")
download.file(url, local_path, mode = "wb")

# Read the downloaded Excel file
data <- read_excel(local_path)
```

Analysis

Data Cleaning

Before diving into the analysis, it's essential to ensure that the dataset is clean. This step involves checking for missing values, handling categorical variables, and splitting the data into training and testing sets for modeling.

```
# Display the first few rows
head(data)
```

```
## # A tibble: 6 x 18
##   RespondentID   Age Gender County Q9c_1 A high tax lev~1 Q9c_2 Having private~2
##          <dbl> <dbl> <chr>  <chr>  <chr>                  <chr>
## 1          469    57 Woman  Vestl~ Absolutely impossible~ Absolutely impossible~
## 2         1126    29 Man    Oslo   Completely agree       Completely agree
## 3          666    46 Man    Rogal~ Partially agree        Partially agree
## 4          672    41 Woman  Nordl~ Partially agree        Absolutely impossible~
## 5          885    50 Man    Trønd~ Completely agree       Completely agree
## 6         1778    22 Woman  Rogal~ Partially agree        Partially agree
## # i abbreviated names:
## #   1: 'Q9c_1 A high tax level is necessary to maintain important public services',
## #   2: 'Q9c_2 Having private schools or hospitals is good. This allows those who want to receive bet
## # i 12 more variables: 'Q9c_3 I prefer to live a quiet life.' <chr>,
## #   'Q9c_4 I wouldn't mind a boring job, as long as it pays well.' <chr>,
## #   'Q9c_5 I like to try everything that gives me a richer inner life.' <chr>,
## #   'Q9c_6 There is too much government intervention and regulation in today's society.' <chr>, ...
```

```r
# Summary of data
summary(data)
```

```
##   RespondentID          Age            Gender             County
##  Min.   :   2.0   Min.   :18.00   Length:1628        Length:1628
##  1st Qu.: 436.8   1st Qu.:33.00   Class :character   Class :character
##  Median : 896.5   Median :46.00   Mode  :character   Mode  :character
##  Mean   :1033.7   Mean   :47.64
##  3rd Qu.:1701.2   3rd Qu.:62.00
##  Max.   :2249.0   Max.   :91.00
##  Q9c_1 A high tax level is necessary to maintain important public services
##  Length:1628
##  Class :character
##  Mode  :character
##
##
##
##  Q9c_2 Having private schools or hospitals is good. This allows those who want to receive better edu
##  Length:1628
##  Class :character
##  Mode  :character
##
##
##
##  Q9c_3 I prefer to live a quiet life.
##  Length:1628
##  Class :character
##  Mode  :character
##
##
##
##  Q9c_4 I wouldn't mind a boring job, as long as it pays well.
##  Length:1628
##  Class :character
##  Mode  :character
##
##
##
##  Q9c_5 I like to try everything that gives me a richer inner life.
##  Length:1628
##  Class :character
##  Mode  :character
##
##
##
##  Q9c_6 There is too much government intervention and regulation in today's society.
##  Length:1628
##  Class :character
##  Mode  :character
##
##
##
##  Q9c_7 We should solve the problems in our own country before spending money to help people in other
```

```
##  Length:1628
##  Class :character
##  Mode  :character
##
##
##
##  Q9c_8 I lack some material goods to live the way I wish.
##  Length:1628
##  Class :character
##  Mode  :character
##
##
##
##  Q9c_9 Life in the countryside is more satisfying than life in cities.
##  Length:1628
##  Class :character
##  Mode  :character
##
##
##
##  Q9c_10 I am constantly looking for new emotional experiences.
##  Length:1628
##  Class :character
##  Mode  :character
##
##
##
##  Q9c_11 The old and well-tested is usually better than brand new inventions.
##  Length:1628
##  Class :character
##  Mode  :character
##
##
##
##  Q9c_12 The worst thing is people who can't fit in with the majority.
##  Length:1628
##  Class :character
##  Mode  :character
##
##
##
##    SampWeight       Segment
##  Min.   :0.0000   Length:1628
##  1st Qu.:0.7963   Class :character
##  Median :0.9728   Mode  :character
##  Mean   :1.0000
##  3rd Qu.:1.0431
##  Max.   :2.1130
```

```r
# how many respondents
length(unique(data$RespondentID))
```

```
## [1] 1628
```

```r
# see unique responses for one q
unique(data$`Q9c_1 A high tax level is necessary to maintain important public services`)
```

```
## [1] "Absolutely impossible to answer" "Completely agree"
## [3] "Partially agree"                 "Completely disagree"
## [5] "Partially disagree"
```

```r
# Convert Segment to factor and inspect it
data$Segment <- as.factor(data$Segment)
table(data$Segment)
```

```
##
##    Down-to-earth       Headstrong            Pious    Retrospective
##              386              277              164              317
##        Searching Well-intentioned
##              243              241
```

```r
# splitting the data
set.seed(123)
splitIndex <- createDataPartition(data$Segment, p = 0.8, list = TRUE, times = 1)
train_data <- data[splitIndex$Resample1, ]
test_data <- data[-splitIndex$Resample1, ]
```

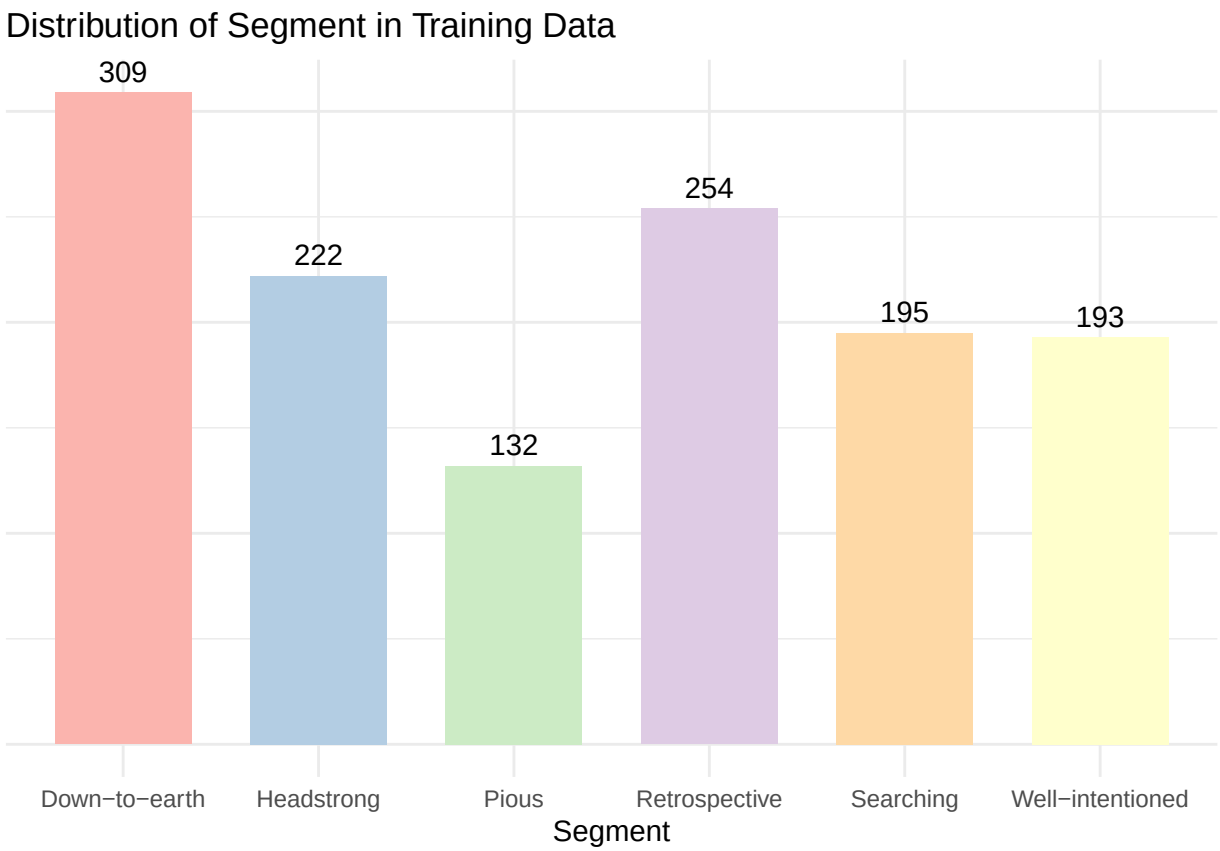
Now we have split the dataset into a train and a test dataset. We have taken a look at the data and confirmed that the questions are categorical as well as segment and county. Age, respondentID is numerical. Age is ranging from 18 to 91. And we have 1628 unique respondents in this dataset.

Data Exploration and Visualization

Understanding the distribution of our target variable and the features is crucial. Let's visualize the distribution of the Segment column and the distribution of responses for some of the questions.

Distribution of the Segment column

```r
ggplot(train_data, aes(x = Segment, fill = Segment)) + geom_bar(width = 0.7) + geom_text(stat = "count"
    aes(label = ..count..), vjust = -0.5) + scale_fill_brewer(palette = "Pastel1") +
    theme_minimal() + labs(title = "Distribution of Segment in Training Data", x = "Segment",
    y = NULL) + theme(legend.position = "none", axis.title.y = element_blank(), axis.text.y = element_bl
    axis.ticks.y = element_blank())
```



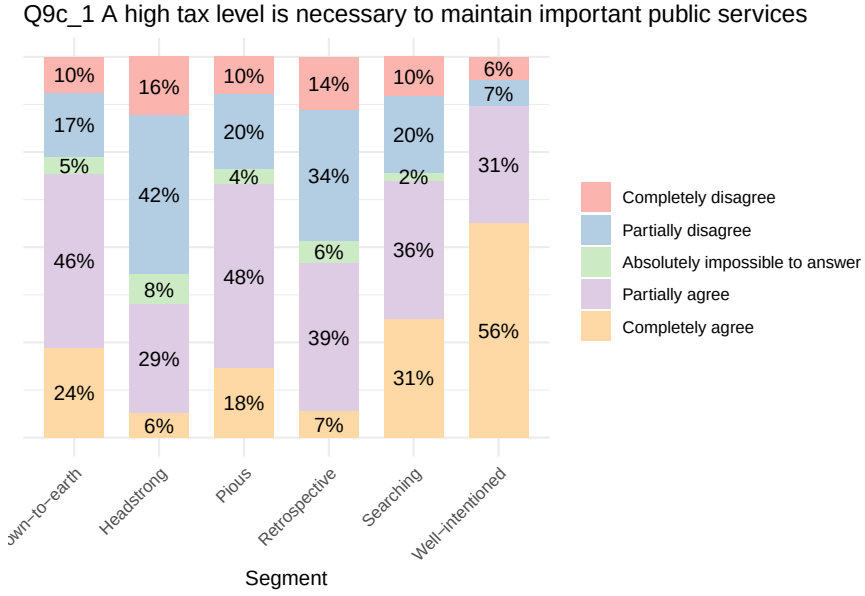
We can see that there are large differences in how many respondents we find in each segment. Ideally they should be more evenly distributed.

Distribution of responses to the questions

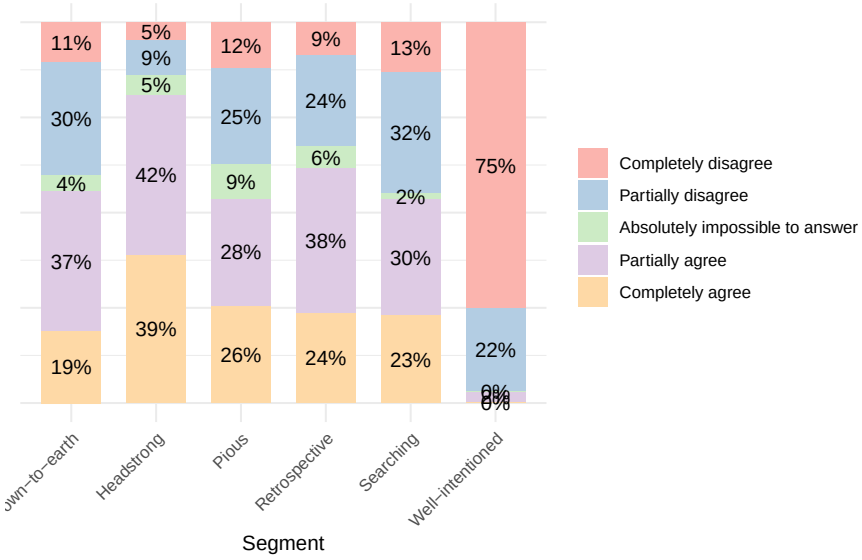
I wanted to see how the different segments respondet to the questions.

```
# First, make sure we use the original dataset's column names
question_cols_original <- colnames(data)[5:16]

# Reshape the data to long format with original column names
long_data_original <- data %>%
    gather(key = "Question", value = "Response", 5:16)

# Create a data frame with percentages
long_data_percent <- long_data_original %>%
    group_by(Question, Segment, Response) %>%
    summarise(Count = n()) %>%
    mutate(Total = sum(Count), Percentage = (Count/Total) * 100) %>%
    ungroup()

# Define the desired order
desired_order <- c("Completely disagree", "Partially disagree", "Absolutely impossible to answer",
    "Partially agree", "Completely agree")
```

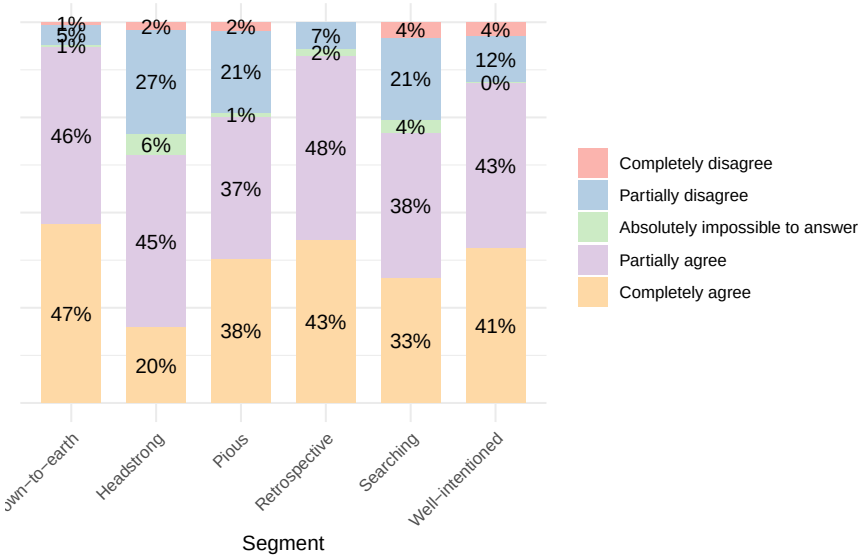
```r
# Plotting function with blank legend title
plot_question_percent <- function(data, question_name) {

    filtered_data <- data %>%
        filter(Question == question_name)

    ggplot(filtered_data, aes(x = Segment, y = Percentage, fill = factor(Response,
        levels = desired_order))) + geom_bar(stat = "identity", width = 0.7, position = "fill") +
        geom_text(aes(label = sprintf("%.0f%%", Percentage)), position = position_fill(vjust = 0.5)) +
        scale_y_continuous(labels = scales::percent_format(scale = 1)) + scale_fill_brewer(palette = "Pa
        name = "") + scale_x_discrete(limits = unique(data$Segment)) + theme_minimal() +
        labs(title = question_name, x = "Segment", y = NULL) + theme(axis.text.x = element_text(angle =
        hjust = 1), axis.title.y = element_blank(), axis.text.y = element_blank(),
        axis.ticks.y = element_blank())
}

# Loop through unique questions in the long data and plot them
for (question_name in question_cols_original) {
    print(plot_question_percent(long_data_percent, question_name))
}
```



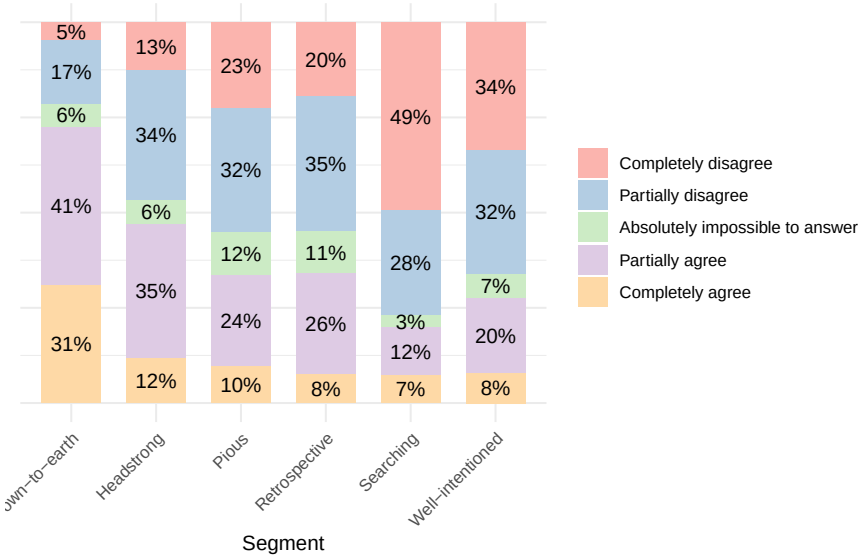
Q9c_2 Having private schools or hospitals is good. This allows those who want t



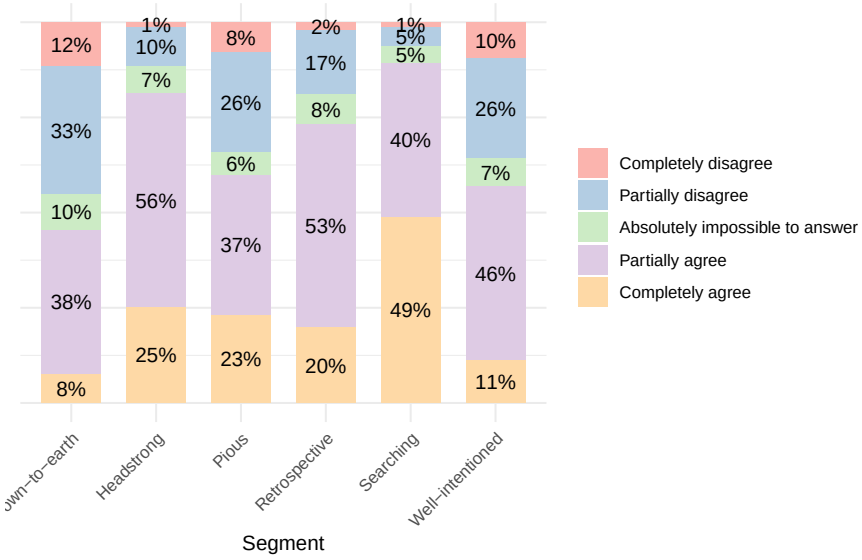
Q9c_3 I prefer to live a quiet life.



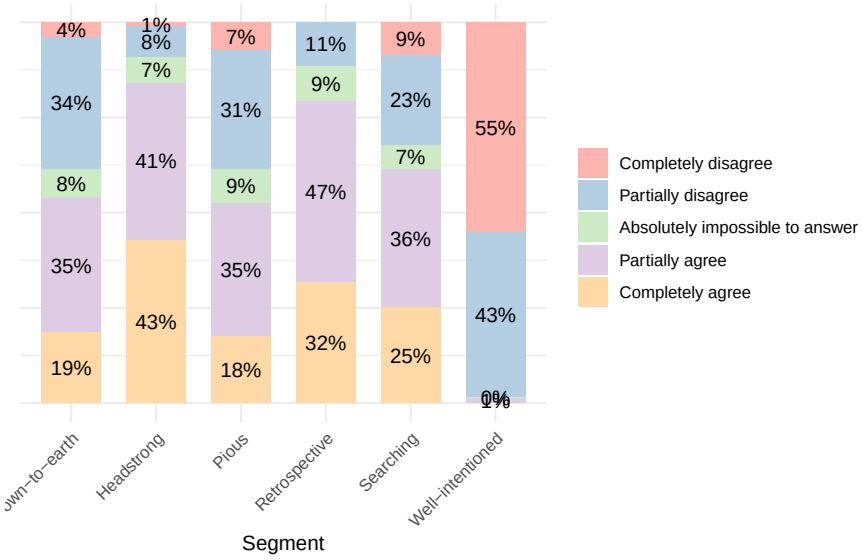
Q9c_4 I wouldn't mind a boring job, as long as it pays well.



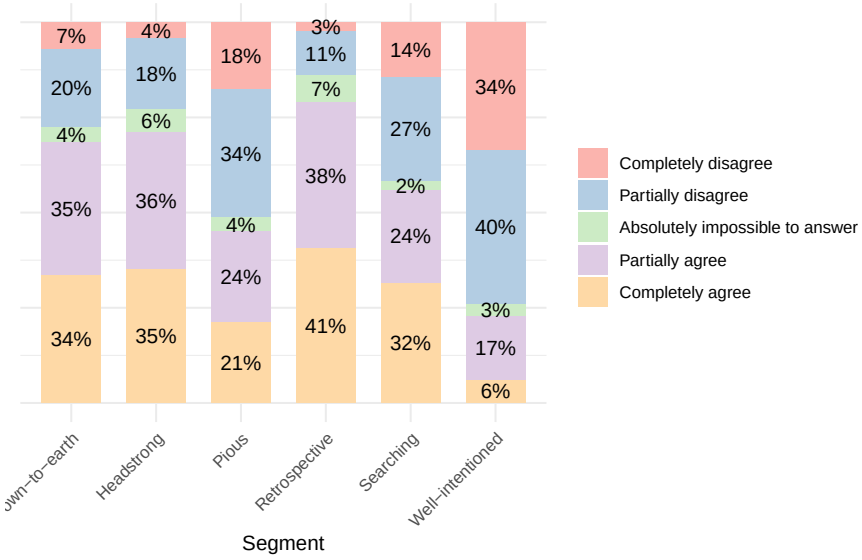
Q9c_5 I like to try everything that gives me a richer inner life.



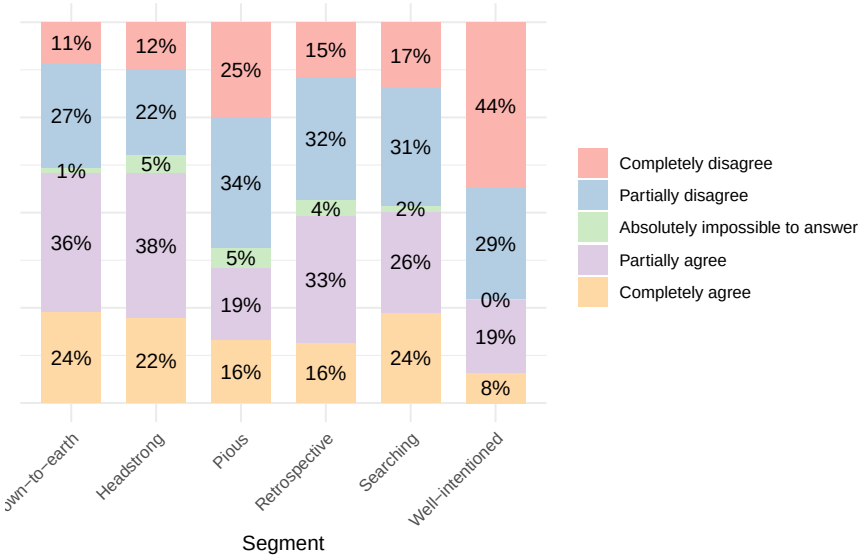
Q9c_6 There is too much government intervention and regulation in today's soci



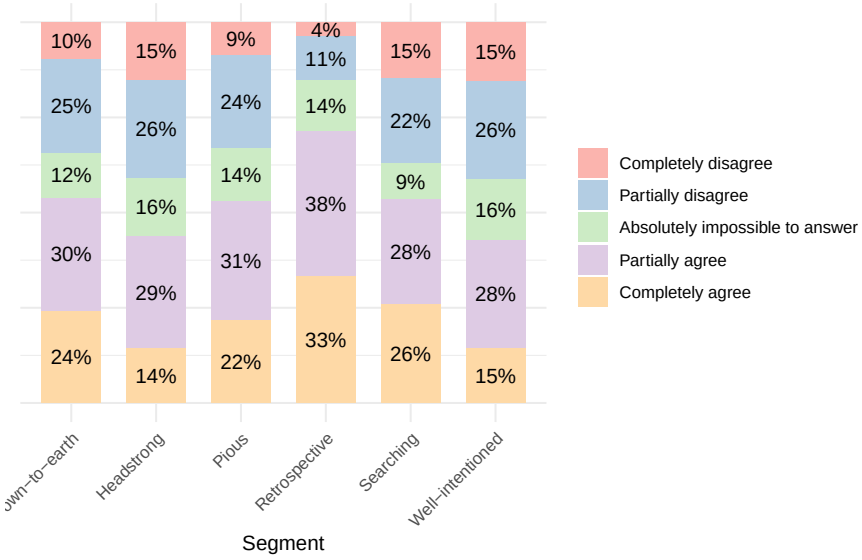
Q9c_7 We should solve the problems in our own country before spending money



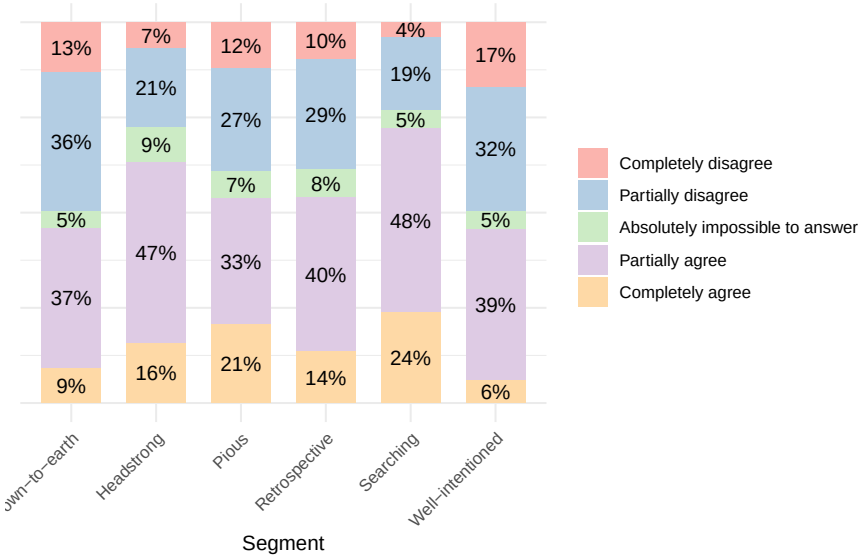
Q9c_8 I lack some material goods to live the way I wish.



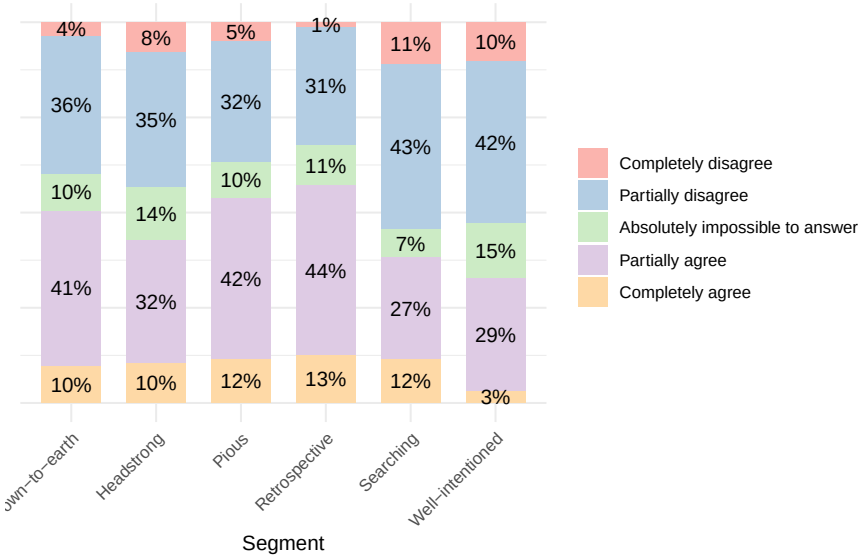
Q9c_9 Life in the countryside is more satisfying than life in cities.

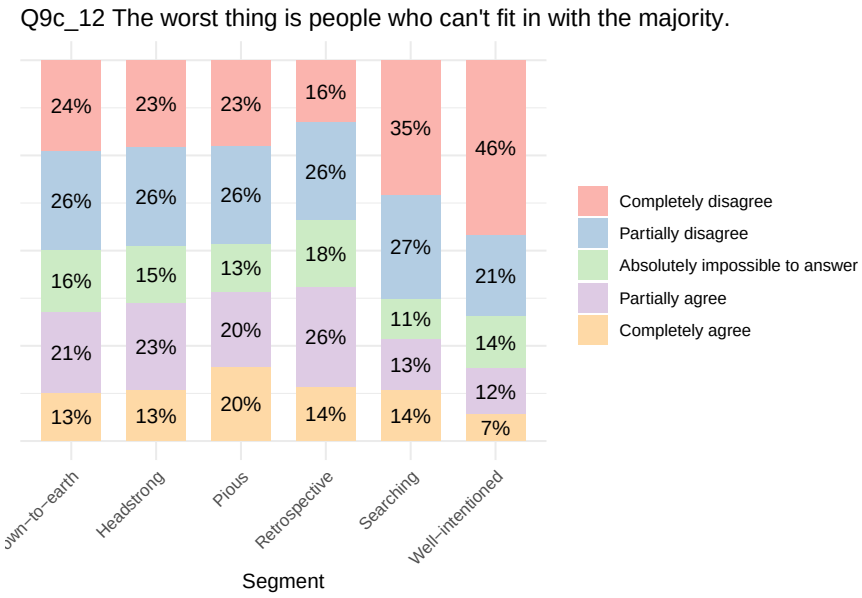


Q9c_10 I am constantly looking for new emotional experiences.



Q9c_11 The old and well−tested is usually better than brand new inventions.

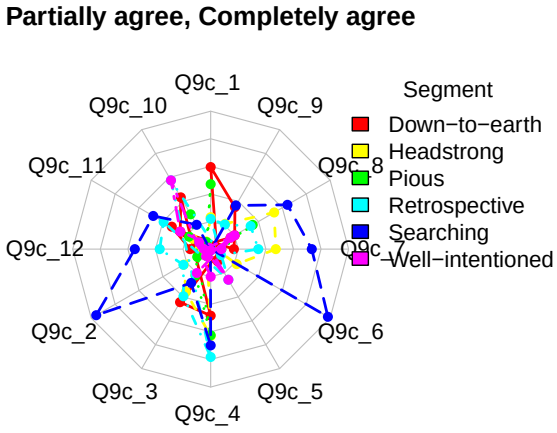




Interesting to see how polarized the "Well-intentioned" answered in a lot of the questions. One might think that it was because of a low number in this group but it is the around the average number.

```
# Load necessary libraries
library(tidyverse)
library(fmsb)
library(gridExtra)
library(grid)


# Aggregate data for spider chart
spider_data <- long_data_percent %>%
    filter(Response %in% c("Partially agree", "Completely agree")) %>%
    group_by(Segment, Question) %>%
    summarise(Total_Percentage = sum(Percentage)) %>%
    spread(key = Question, value = Total_Percentage)

# Store the full question names for the table
full_question_names <- data.frame(Question = colnames(spider_data)[-1])

# Truncate question names to the first 6 letters for the spider chart
colnames(spider_data)[-1] <- substr(colnames(spider_data)[-1], 1, 6)

# Prepare data for spider chart
max_values <- apply(spider_data[, -1], 2, max)
min_values <- rep(0, length(max_values))

# Set the segments as row names
row.names(spider_data) <- spider_data$Segment

# Bind the min and max values, and remove the Segment column
spider_data_final <- rbind(min_values, spider_data[, -1], max_values)

# Define colors for each segment
```

```r
colors <- rainbow(n = nrow(spider_data))



# Create the spider chart with defined colors
spider_plot <- radarchart(spider_data_final, pcol = colors, plwd = 2, cglcol = "grey",
    cglty = 1, axislabcol = "grey", caxislabels = seq(0, 100, 20), cglwd = 0.8, title = "Partially agree
    show.legend = FALSE)


# Create the table with full question names
table_plot <- tableGrob(full_question_names, theme = ttheme_default(base_size = 8,
    core = list(fg_params = list(hjust = 0, x = 0))))
# Create a custom legend and grab it as a grob
legend("topright", legend = spider_data$Segment, fill = colors, title = "Segment",
    bty = "n")
legend_grob <- grid::grid.grab()
```




```r
# Arrange the spider chart, custom legend, and table side by side
grid.arrange(spider_plot, legend_grob, table_plot, ncol = 3)
```

	Question
1	Q9c_1 A high tax level is necessary to maintain important public services
2	Q9c_10 I am constantly looking for new emotional experiences.
3	Q9c_11 The old and well−tested is usually better than brand new inventions.
4	Q9c_12 The worst thing is people who can't fit in with the majority.
5	Q9c_2 Having private schools or hospitals is good. This allows those who want to receive bet
6	Q9c_3 I prefer to live a quiet life.
7	Q9c_4 I wouldn't mind a boring job, as long as it pays well.
8	Q9c_5 I like to try everything that gives me a richer inner life.
9	Q9c_6 There is too much government intervention and regulation in today's society.
10	Q9c_7 We should solve the problems in our own country before spending money to help peo
11	Q9c_8 I lack some material goods to live the way I wish.
12	Q9c_9 Life in the countryside is more satisfying than life in cities.

A bit of a tricky chart but I wanted to see how the segments, while only looking at teir positive response, respondent compared to each other.

Modeling Approach

For this analysis, we've chosen Random Forest, kNN, and Naive Bayes. Here's why:

Random Forest:

Given the dataset's mix of categorical responses to different questions, the Random Forest model becomes an optimal choice. This ensemble method thrives in scenarios with both numerical and categorical features. Its parallel processing ability accelerates the training process, making it both efficient and accurate in prediction.

k-Nearest Neighbors (kNN):

The kNN algorithm was chosen for its expertise in classification tasks. Considering the categorical nature of the dataset, I introduced dummy variables. This step was vital to ensure the kNN could process numerical data and prevent unintended ordinal interpretations of the categorical responses.

Naive Bayes:

The dataset consists of multiple categorical responses, leading to a high-dimensional space when considering each unique response. Naive Bayes, with its probabilistic approach, is adept at handling such high-dimensional datasets. It computes the likelihood of each category, making it particularly suited for our dataset's nature.

The dataset was split into a training set (80%) and a testing set (20%). This split ensures that we have a sufficient amount of data for training our models while also retaining an adequate portion for validation.

Now, we'll transition to the Results section, where we'll fit our models on the training data and evaluate their performance on the test data.

Results

```
# Ensure column names are the simplified versions
colnames(train_data)[5:16] <- paste("Q", 1:12, sep = "")
colnames(test_data)[5:16] <- paste("Q", 1:12, sep = "")

# Construct the formula with the new column names
model_formula <- as.formula(paste("Segment ~", paste(paste("Q", 1:12, sep = ""),
    collapse = " + ")))

# Random Forest model
set.seed(123)
rf_model <- randomForest(model_formula, data = train_data, importance = TRUE, ntree = 100)

# Predict using the Random Forest model
rf_predictions <- predict(rf_model, test_data)

# Convert character columns to factors
train_data[, 5:16] <- lapply(train_data[, 5:16], factor)
test_data[, 5:16] <- lapply(test_data[, 5:16], factor)
# Create the dummy variables again
```

```r
train_data_dummies <- model.matrix(~. - 1, data = train_data[, 5:16])
test_data_dummies <- model.matrix(~. - 1, data = test_data[, 5:16])

# Add the Segment column back to the datasets
train_data_dummies <- as.data.frame(train_data_dummies)
train_data_dummies$Segment <- train_data$Segment

test_data_dummies <- as.data.frame(test_data_dummies)
test_data_dummies$Segment <- test_data$Segment

# k-Nearest Neighbors model with dummy variables
set.seed(123)
knn_predictions <- kknn(Segment ~ ., train_data_dummies, test_data_dummies, k = 5,
    distance = 1, kernel = "optimal")


# Naive Bayes model
set.seed(123)
nb_model <- naiveBayes(model_formula, data = train_data)
nb_predictions <- predict(nb_model, test_data[, 5:16])

# Evaluate model performance
rf_accuracy <- sum(diag(confusionMatrix(rf_predictions, test_data$Segment)$table))/sum(confusionMatrix(
    test_data$Segment)$table)
knn_accuracy <- sum(diag(confusionMatrix(knn_predictions$fit, test_data$Segment)$table))/sum(confusionMa
    test_data$Segment)$table)
nb_accuracy <- sum(diag(confusionMatrix(nb_predictions, test_data$Segment)$table))/sum(confusionMatrix(
    test_data$Segment)$table)


# Create the data frame
results_df <- data.frame(Model = c("Random Forest", "k-Nearest Neighbors", "Naive Bayes"),
    Accuracy = sprintf("%.2f%%", c(rf_accuracy, knn_accuracy, nb_accuracy) * 100)  # Convert to percent
)

# Display it as a table
results_df %>%
    gt() %>%
    tab_header(title = "Model Accuracy")
```

Model Accuracy

Model	Accuracy
Random Forest	51.08%
k-Nearest Neighbors	41.18%
Naive Bayes	53.87%

The table above summarizes the accuracy of our three models. As we can observe:

- The Random Forest model achieved an accuracy of 51.08%.
- The k-Nearest Neighbors model achieved an accuracy of 41.18%.

- The Naive Bayes model achieved an accuracy of 53.87%.

This provides a comparative understanding of how each model performs on our dataset. The accuracy metric is chosen as it gives a straightforward understanding of how often the model predicts correctly.

I think it could be interesting to see if by adding variables that are usually always included in questioners can impact the models, preferably increase the accuracy. In this dataset we have Age, Gender and county we can add.

(For code please see rmd or script file. Basically the same as previous except including the extra variables)

Combining the two results:

```
# Create the data frames
results_df <- data.frame(Model = c("Random Forest", "k-Nearest Neighbors", "Naive Bayes"),
    Accuracy_Standard = sprintf("%.2f%%", c(rf_accuracy, knn_accuracy, nb_accuracy) *
        100)  # Convert to percentage with a % sign
)

results_df_extended <- data.frame(Model = c("Random Forest", "k-Nearest Neighbors",
    "Naive Bayes"), Accuracy_Extended = sprintf("%.2f%%", c(rf_accuracy_extended,
    knn_accuracy_extended, nb_accuracy_extended) * 100)  # Convert to percentage with a % sign
)
# Merge the data frames
combined_df <- merge(results_df, results_df_extended, by = "Model", all = TRUE, sort = FALSE)

# Reorder the Model column to maintain desired order
desired_order <- c("Random Forest", "k-Nearest Neighbors", "Naive Bayes", "Random Forest (Extended)",
    "k-Nearest Neighbors (Extended)", "Naive Bayes (Extended)")
combined_df$Model <- factor(combined_df$Model, levels = desired_order)

# Display the combined results in a table
combined_df %>%
    gt() %>%
    tab_header(title = "Model Accuracy Comparison")
```

Model Accuracy Comparison

Model	Accuracy_Standard	Accuracy_Extended
Random Forest	51.08%	61.30%
k-Nearest Neighbors	41.18%	32.20%
Naive Bayes	53.87%	51.39%

Conclusion

Throughout this analysis, we've employed various models to predict the segment of individuals based on their responses to a set of opinion-based questions.

The integration of background variables, specifically Age, Gender, and County, has had a mixed impact on our model accuracies:

Random Forest:

This model experienced a notable improvement with an increase of approximately 10 percentage points in accuracy. The addition of the demographic variables possibly provided the Random Forest algorithm with more information to make refined splits in the decision trees, enhancing its predictive capabilities.

k-Nearest Neighbors (kNN):

The accuracy for kNN decreased by nearly 9 percentage points upon the inclusion of the extra variables. This suggests that the addition of these variables introduced more complexity and noise into the distance calculations that kNN relies on. It's a reminder that kNN can be sensitive to the scale and nature of the data, and not all additional features will necessarily enhance its performance.

Naive Bayes:

The performance of the Naive Bayes model showed a slight dip (around 2.5 percentage points) with the addition of the background variables. This indicates that while these variables might be informative, their inclusion in the model might have somewhat complicated the conditional probability estimations.

In summary, while one model benefited from the richer dataset, others struggled with the added complexity. This underscores the importance of thoughtful feature selection and the need to continually validate and test models, especially as more variables are incorporated. It's evident that not all additional features guarantee better performance across all algorithms. It's crucial to understand the nuances and sensitivities of each modeling approach to make informed decisions about feature integration. Our aim of exceeding 70% accuracy was not met. So the company needs to consider the cost of including 30 background questions into questionnaires that gives an accuarcy of 70% vs including 11 questions and three basic background questions and then obtaining an accuracy of 61.30%. 70% vs 61.30% is not that far apart in mind.

Potential Impact:

Achieving sub optimal accuracy can lead to misguided analyses of the segments. For a public broadcaster like this company, such inaccuracies could have adverse consequences. Misinterpreting which programs various segments prefer may result in the broadcaster presenting unsuitable content to the wrong audience. This mismatch could diminish viewer satisfaction and engagement.

Limitations:

Our models rely heavily on the quality and number of respondents in the data set. If the dataset isn't representative of the broader population, our predictions might not generalize well outside this sample. The number of respondents might be to few to make this predictions, but again it comes down to cost of having more respondents vs the accuracy level we get from this dataset. Including more background variables, like political party, education, status etc might impact the models.

Future Work:

1) Model Optimization: run several models with tuning to see if we achive better accuracy.
2) Incorporating Additional Data: Including demographic or contextual details can provide richer insights and improve model accuracy.
3) Complex Models: Exploring deep learning techniques or an ensemble of different models might yield better performance.

References

Primary Literature and Textbooks

- Irizarry, R. A. (n.d.). Introduction to Data Science. Retrieved from http://rafalab.dfci.harvard.edu/dsbook/
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.
- Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Software and Libraries

- R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
- Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22. [For the randomForest package in R]
- Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). kernlab – An S4 Package for Kernel Methods in R. Journal of Statistical Software, 11(9), 1-20. [For the kknn function]