Implementing a Decision Tree Model and Multinomial Logistic Model

Le' Sean Roberts (NYC Office of Management and Budget College Aide)

Summary

This development investigates the application of decision models and multinomial logistic regression in towards applying school districts or primary program type as target or label based on participant statistics. The features applied generally may be considered as neither having strong social attributes nor strong economic attributes, however, the data set is applied to provide a view of the implementation of the prior mentioned chosen machine learning models.

Introduction

Special education programs play a crucial role in addressing the diverse learning needs of students with disabilities or learning difficulties. However, determining the most appropriate educational interventions for individual students can be challenging due to the complex array of factors involved. Decision models and statistical techniques offer valuable tools for guiding these decisions by leveraging student data and identifying patterns that inform program recommendations. In this development, we explore the application of decision models and multinomial logistic regression specifically in the context of participant data, school regions and program types.

The Data Cleaning Process

```
library(readr)
Students_Receiving_Recommended_Special_Education_Programs <-
  read_csv("C:/Users/verlene/Downloads/Students_Receiving_Recommended_Special_Education_Pr
library(tidyverse)
```

```r
glimpse(Students_Receiving_Recommended_Special_Education_Programs)
```

```
Rows: 97
Columns: 8
$ `School District`             <dbl> 1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5~
$ `Primary Program Type`        <chr> "Integrated Co-Teaching Services", "SETS~
$ `Fully Receiving`             <dbl> 749, 21, 135, 4482, 259, 943, 1163, 86, ~
$ `Percent Fully Receiving`     <dbl> 0.42, 0.13, 0.29, 0.55, 0.24, 0.52, 0.47~
$ `Partially Receiving`         <chr> "626", "109", "227", "2821", "548", "749~
$ `Percent Partially Receiving` <chr> "36%", "66%", "48%", "35%", "51%", "41%"~
$ `Not Receiving`               <chr> "388", "34", "107", "831", "269", "122",~
$ `Percent Not Receiving`       <chr> "22%", "21%", "23%", "10%", "25%", "7%",~
```

```r
# Showing first 6 rows.
head(Students_Receiving_Recommended_Special_Education_Programs)
```

```
# A tibble: 6 x 8
  `School District` `Primary Program Type`          `Fully Receiving`
              <dbl> <chr>                                       <dbl>
1                 1 Integrated Co-Teaching Services               749
2                 1 SETSS                                          21
3                 1 Special Class                                 135
4                 2 Integrated Co-Teaching Services              4482
5                 2 SETSS                                         259
6                 2 Special Class                                 943
# i 5 more variables: `Percent Fully Receiving` <dbl>,
#   `Partially Receiving` <chr>, `Percent Partially Receiving` <chr>,
#   `Not Receiving` <chr>, `Percent Not Receiving` <chr>
```

```r
# Dropping rows with impractical instances. Namely, rows 12 and 96.
Students_Receiving_Recommended_Special_Education_Programs <-
  Students_Receiving_Recommended_Special_Education_Programs[-c(12, 96), ]
```

```r
# Dropping any possible rows with NAs.
Students_Receiving_Recommended_Special_Education_Programs <-
  Students_Receiving_Recommended_Special_Education_Programs |>
  na.omit()
```

```r
dim(Students_Receiving_Recommended_Special_Education_Programs)
```

```
[1] 94  8
```

```r
# Identify unique instances for each column
unique_instances <-
  lapply(Students_Receiving_Recommended_Special_Education_Programs,
         unique)

# Print unique instances for each column
print(unique_instances)
```

```
$`School District`
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[26] 26 27 28 29 30 31 32

$`Primary Program Type`
[1] "Integrated Co-Teaching Services" "SETSS"
[3] "Special Class"

$`Fully Receiving`
 [1]  749   21  135 4482  259  943 1163   86  395  987   71  286   68  822 1175
[16]   54  403 1603  115 1082 1979  131 1290 1550  150 1040 2925  305 2376 2160
[31]  155 2225 1575  349 2384  880  697  997  215  909 2265  153 1106  389   49
[46]  224  721   89  491  551   74  430 1389  117 1089 2711  406 1419 1900  175
[61] 1591 2133   91 1533  450   82  632 2827  151 2745 1987   85 1110 1692  144
[76] 1899 3010  299 1924 2187  196 1415 1062   93  912 1709  108  829 4641  229
[91] 3602  649   50

$`Percent Fully Receiving`
 [1] 0.42 0.13 0.29 0.55 0.24 0.52 0.47 0.16 0.50 0.26 0.18 0.49 0.51 0.15 0.41
[16] 0.61 0.63 0.25 0.57 0.23 0.45 0.56 0.31 0.54 0.19 0.53 0.72 0.76 0.21 0.48
[31] 0.28 0.60 0.32 0.35 0.39 0.36 0.62 0.59 0.33 0.58 0.78 0.22 0.30 0.66 0.73
[46] 0.67 0.46

$`Partially Receiving`
 [1] "626"  "109"  "227"  "2821" "548"  "749"  "922"  "343"  "310"  "678"
[11] "81"   "616"  "208"  "724"  "879"  "178"  "461"  "840"  "113"  "733"
[21] "861"  "284"  "772"  "1223" "323"  "1139" "1831" "463"  "1555" "1434"
[31] "483"  "1843" "506"  "171"  "723"  "694"  "183"  "625"  "800"  "165"
```

```
[41] "377"  "1842" "268"  "599"  "304"  "90"   "164"  "1077" "255"  "667"
[51] "103"  "246"  "959"  "180"  "642"  "1313" "199"  "869"  "1257" "217"
[61] "968"  "730"  "155"  "420"  "458"  "99"   "356"  "1633" "339"  "1852"
[71] "974"  "130"  "685"  "705"  "1449" "1075" "119"  "662"  "987"  "160"
[81] "1221" "602"  "97"   "400"  "1222" "215"  "528"  "3217" "325"  "2167"
[91] "464"  "100"

$`Percent Partially Receiving`
 [1] "36%" "66%" "48%" "35%" "51%" "41%" "38%" "64%" "39%" "56%" "55%" "43%"
[13] "46%" "32%" "37%" "27%" "34%" "49%" "47%" "60%" "44%" "23%" "53%" "25%"
[25] "40%" "50%" "58%" "52%" "54%" "33%" "30%" "26%" "24%" "21%" "31%" "45%"
[37] "42%" "28%"

$`Not Receiving`
 [1] "388" "34"  "107" "831" "269" "122" "369" "111" "85"  "221" "18"  "200"
[13] "105" "123" "266" "120" "131" "172" "8"   "147" "301" "102" "217" "422"
[25] "190" "135" "457" "210" "401" "163" "146" "109" "103" "45"  "138" "89"
[37] "44"  "267" "25"  "214" "476" "119" "144" "106" "16"  "17"  "265" "121"
[49] "54"  "139" "15"  "73"  "285" "29"  "69"  "373" "153" "378" "186" "118"
[61] "40"  "22"  "158" "27"  "441" "196" "645" "66"  "166" "293" "281" "98"
[73] "38"  "41"  "112" "74"  "95"  "321" "39"  "129" "318" "99"  "397" "86"
[85] "148" "21"

$`Percent Not Receiving`
 [1] "22%" "21%" "23%" "10%" "25%" "7%"  "15%" "11%" "12%" "18%" "28%" "34%"
[13] "13%" "3%"  "20%" "29%" "6%"  "9%"  "5%"  "17%" "1%"  "8%"  "26%" "14%"
[25] "4%"  "2%"  "16%"
```

Data Wrangling and Feature Engineering

```r
# Changing Primary Program Type column to have categorical values.
# first, converting "Primary Program Type" column to a factor.
Students_Receiving_Recommended_Special_Education_Programs$`Primary Program Type` <-
  as.factor(
    Students_Receiving_Recommended_Special_Education_Programs$`Primary Program Type`)

# Convert factor levels to numeric values
Students_Receiving_Recommended_Special_Education_Programs$`Primary Program Type` <-
  as.numeric(
    Students_Receiving_Recommended_Special_Education_Programs$`Primary Program Type`)
```

```r
# Print the updated data frame
print(Students_Receiving_Recommended_Special_Education_Programs)
```

```
# A tibble: 94 x 8
   `School District` `Primary Program Type` `Fully Receiving`
               <dbl>                  <dbl>             <dbl>
 1                 1                      1               749
 2                 1                      2                21
 3                 1                      3               135
 4                 2                      1              4482
 5                 2                      2               259
 6                 2                      3               943
 7                 3                      1              1163
 8                 3                      2                86
 9                 3                      3               395
10                 4                      1               987
# i 84 more rows
# i 5 more variables: `Percent Fully Receiving` <dbl>,
#   `Partially Receiving` <chr>, `Percent Partially Receiving` <chr>,
#   `Not Receiving` <chr>, `Percent Not Receiving` <chr>
```

```r
# Convert "Partially Receiving" column to double
Students_Receiving_Recommended_Special_Education_Programs$`Partially Receiving` <-
  as.numeric(Students_Receiving_Recommended_Special_Education_Programs$`Partially Receivin

# Convert "Percent Partially Receiving" column to double
Students_Receiving_Recommended_Special_Education_Programs$`Percent Partially Receiving` <-
  as.numeric(gsub("%", "",
                  Students_Receiving_Recommended_Special_Education_Programs$`Percent Parti

# Print the updated structure of the dataframe
str(Students_Receiving_Recommended_Special_Education_Programs)
```

```
tibble [94 x 8] (S3: tbl_df/tbl/data.frame)
 $ School District            : num [1:94] 1 1 1 2 2 2 3 3 3 4 ...
 $ Primary Program Type       : num [1:94] 1 2 3 1 2 3 1 2 3 1 ...
 $ Fully Receiving            : num [1:94] 749 21 135 4482 259 ...
 $ Percent Fully Receiving    : num [1:94] 0.42 0.13 0.29 0.55 0.24 0.52 0.47 0.16 0.5 0.52
 $ Partially Receiving        : num [1:94] 626 109 227 2821 548 ...
 $ Percent Partially Receiving: num [1:94] 36 66 48 35 51 41 38 64 39 36 ...
```

```
 $ Not Receiving              : chr [1:94] "388" "34" "107" "831" ...
 $ Percent Not Receiving      : chr [1:94] "22%" "21%" "23%" "10%" ...
 - attr(*, "na.action")= 'omit' Named int 95
  ..- attr(*, "names")= chr "95"
```

```
# Convert "Not Receiving" column to double
Students_Receiving_Recommended_Special_Education_Programs$`Not Receiving` <-
  as.numeric(Students_Receiving_Recommended_Special_Education_Programs$`Not Receiving`)

# Convert "Percent Not Receiving" column to double
Students_Receiving_Recommended_Special_Education_Programs$`Percent Not Receiving` <-
  as.numeric(gsub("%", "",
                  Students_Receiving_Recommended_Special_Education_Programs$`Percent Not R

# Print the updated structure of the dataframe
str(Students_Receiving_Recommended_Special_Education_Programs)
```

```
tibble [94 x 8] (S3: tbl_df/tbl/data.frame)
 $ School District            : num [1:94] 1 1 1 2 2 2 3 3 3 4 ...
 $ Primary Program Type       : num [1:94] 1 2 3 1 2 3 1 2 3 1 ...
 $ Fully Receiving            : num [1:94] 749 21 135 4482 259 ...
 $ Percent Fully Receiving    : num [1:94] 0.42 0.13 0.29 0.55 0.24 0.52 0.47 0.16 0.5 0.52
 $ Partially Receiving        : num [1:94] 626 109 227 2821 548 ...
 $ Percent Partially Receiving: num [1:94] 36 66 48 35 51 41 38 64 39 36 ...
 $ Not Receiving              : num [1:94] 388 34 107 831 269 122 369 111 85 221 ...
 $ Percent Not Receiving      : num [1:94] 22 21 23 10 25 7 15 21 11 12 ...
 - attr(*, "na.action")= 'omit' Named int 95
  ..- attr(*, "names")= chr "95"
```

```
# Convert "Percent Not Receiving" column to double
Students_Receiving_Recommended_Special_Education_Programs$`Percent Not Receiving` <-
  as.numeric(gsub("%", "",
                  Students_Receiving_Recommended_Special_Education_Programs$`Percent Not R

# Print the updated structure of the dataframe
str(Students_Receiving_Recommended_Special_Education_Programs)
```

```
tibble [94 x 8] (S3: tbl_df/tbl/data.frame)
 $ School District            : num [1:94] 1 1 1 2 2 2 3 3 3 4 ...
 $ Primary Program Type       : num [1:94] 1 2 3 1 2 3 1 2 3 1 ...
```

```
 $ Fully Receiving            : num [1:94] 749 21 135 4482 259 ...
 $ Percent Fully Receiving    : num [1:94] 0.42 0.13 0.29 0.55 0.24 0.52 0.47 0.16 0.5 0.52
 $ Partially Receiving        : num [1:94] 626 109 227 2821 548 ...
 $ Percent Partially Receiving: num [1:94] 36 66 48 35 51 41 38 64 39 36 ...
 $ Not Receiving              : num [1:94] 388 34 107 831 269 122 369 111 85 221 ...
 $ Percent Not Receiving      : num [1:94] 0.22 0.21 0.23 0.1 0.25 0.07 0.15 0.21 0.11 0.12
 - attr(*, "na.action")= 'omit' Named int 95
  ..- attr(*, "names")= chr "95"
```

```
# Convert "Percent Not Receiving" column to double
Students_Receiving_Recommended_Special_Education_Programs$`Percent Partially Receiving` <-
  as.numeric(gsub("%", "",
                  Students_Receiving_Recommended_Special_Education_Programs$`Percent Parti

# Print the updated structure of the dataframe
str(Students_Receiving_Recommended_Special_Education_Programs)
```

```
tibble [94 x 8] (S3: tbl_df/tbl/data.frame)
 $ School District            : num [1:94] 1 1 1 2 2 2 3 3 3 4 ...
 $ Primary Program Type       : num [1:94] 1 2 3 1 2 3 1 2 3 1 ...
 $ Fully Receiving            : num [1:94] 749 21 135 4482 259 ...
 $ Percent Fully Receiving    : num [1:94] 0.42 0.13 0.29 0.55 0.24 0.52 0.47 0.16 0.5 0.52
 $ Partially Receiving        : num [1:94] 626 109 227 2821 548 ...
 $ Percent Partially Receiving: num [1:94] 0.36 0.66 0.48 0.35 0.51 0.41 0.38 0.64 0.39 0.36
 $ Not Receiving              : num [1:94] 388 34 107 831 269 122 369 111 85 221 ...
 $ Percent Not Receiving      : num [1:94] 0.22 0.21 0.23 0.1 0.25 0.07 0.15 0.21 0.11 0.12
 - attr(*, "na.action")= 'omit' Named int 95
  ..- attr(*, "names")= chr "95"
```

Decision Tree Model Development and Evaluation

```
library(tidymodels)
```

```
-- Attaching packages ------------------------------------ tidymodels 1.1.1 --

v broom        1.0.5     v rsample      1.2.0
v dials        1.2.0     v tune         1.1.2
v infer        1.0.5     v workflows    1.1.3
v modeldata    1.3.0     v workflowsets 1.0.1
```

```
v parsnip      1.1.1     v yardstick    1.3.0
v recipes      1.0.9

-- Conflicts ---------------------------------------- tidymodels_conflicts() --
x scales::discard() masks purrr::discard()
x dplyr::filter()   masks stats::filter()
x recipes::fixed()  masks stringr::fixed()
x dplyr::lag()      masks stats::lag()
x yardstick::spec() masks readr::spec()
x recipes::step()   masks stats::step()
* Learn how to get started at https://www.tidymodels.org/start/
```

```r
# Developing train-test split; will be 80-20.

# Set seed for reproducibility
set.seed(123)

# Convert numeric target variable to factor
Students_Receiving_Recommended_Special_Education_Programs$`School District` <- as.factor(S

# Create an index for train-test split
index <- sample(1:nrow(Students_Receiving_Recommended_Special_Education_Programs),
                size = round(0.8 * nrow(Students_Receiving_Recommended_Special_Education_P

# Split the data into training and testing sets
train_data <-
  Students_Receiving_Recommended_Special_Education_Programs[index, ]
test_data <-
  Students_Receiving_Recommended_Special_Education_Programs[-index, ]


# Define the decision tree model specification
tree_spec <- decision_tree() |>
  set_engine("rpart") |>  # Using the rpart engine
  set_mode("classification")  # Specify classification or regression

# Define the model formula
formula <- `School District` ~ .

# Train the decision tree model
tree_fit <- tree_spec |>
  fit(formula, data = train_data)
```

```r
# Make predictions on the testing set
predictions <- predict(tree_fit, new_data = test_data)

# Compute confusion matrix
confusion_matrix <- table(Actual = test_data$`School District`, Predicted = predictions$.p

# Compute Accuracy
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)

print(paste("Accuracy:", accuracy))
```

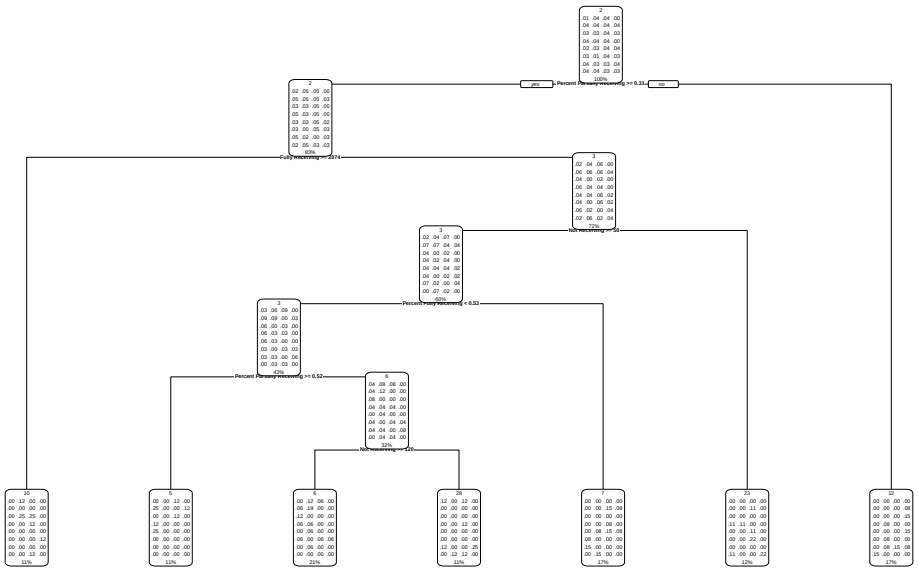
```
[1] "Accuracy: 0.0526315789473684"
```

```r
library(rpart)
library(rpart.plot)

# Train the decision tree model
tree_model <- rpart(`School District` ~ ., data = train_data)

# Plot the decision tree
rpart.plot(tree_model)
```

```
Warning: All boxes will be white (the box.palette argument will be ignored) because
the number of classes in the response 32 is greater than length(box.palette) 6.
To silence this warning use box.palette=0 or trace=-1.
```



Multinomial Logistic Regression Model Development and Evaluation

```
# Step 1: Prepare Data
# Assuming 'data' is your dataframe and 'response' is your outcome variable
data_recipe <- recipe(`School District` ~ .,
                      data = Students_Receiving_Recommended_Special_Education_Programs) |>
  step_normalize(all_numeric())  # Normalize numeric predictors if needed

# Step 2: Define Model
model_spec <- multinom_reg() |>
  set_engine("nnet")  # Specify multinomial logistic regression model engine

# Step 3: Create Workflow
workflow <- workflow() |>
  add_recipe(data_recipe) |>
  add_model(model_spec)

# Step 4: Fit and Evaluate
# Fit the workflow to your data
fit_wf <- workflow |>
  fit(Students_Receiving_Recommended_Special_Education_Programs)
```

```r
# Step 5: Evaluate Performance
results <- fit_wf |>
  predict(Students_Receiving_Recommended_Special_Education_Programs) |>
  bind_cols(Students_Receiving_Recommended_Special_Education_Programs) |>
  metrics(truth = `School District`, estimate = .pred_class)
print(results)
```

```
# A tibble: 2 x 3
  .metric  .estimator .estimate
  <chr>    <chr>          <dbl>
1 accuracy multiclass     0.585
2 kap      multiclass     0.572
```

Notice

In similar fashion to district being the target, "Primary Program Type" can also be made ta feature while having "School District" as a feature among the other features.

Conclusion

This study sheds light on the application of decision models and multinomial logistic regression in towards special education programs data. By leveraging student data and statistical techniques, educators and policymakers can possibly make more informed and personalized recommendations that meet the diverse needs of learners. The findings of this research contribute to the ongoing efforts to improve the effectiveness and efficiency of special education services, ultimately enhancing educational outcomes and promoting inclusivity in schools. Moving forward, continued research and collaboration are essential to further refine these models and ensure equitable access to quality education for all students.

References

Education, N. D. of. (2021, February 11). *Students receiving recommended special education programs by program type: NYC open data.* Students Receiving Recommended Special Education Programs by Program Type | NYC Open Data. https://data.cityofnewyork.us/Education/Students-Receiving-Recommended-Special-Education-P/6thv-9wgt/about_data