HarvardX Professional Certificate in Data Science PH125.9x: Capstone Project_Choose Your Own!

Tan Wei Lun

2025-06-24

1. Introduction

This project uses the AI Tools Usage Among Global High School Students dataset (downloaded from Kaggle), a fully synthetic simulation of 500 students worldwide and their academic use of AI tools in 2025. No personal or survey data were collected and every record was generated via probabilistic logic to capture realistic patterns in demographics (age, gender, country, grade), binary adoption flags for major AI tools (ChatGPT, Gemini, Grammarly, QuillBot, Notion AI, Phind, EduChat, Other), and conditional usefulness ratings. The analysis proceeds in two stages: first, a binary classifier to predict whether a student uses any AI tool; and second, a multiclass model to predict which specific tool an adopter chooses. Model performance for both stages is evaluated on a hold-out test set using ROC-AUC, accuracy, and F1-score to assess predictive accuracy and generalizability.

2. Data Preparation

In preparing the data, we first loaded the CSV file int R. We fixed the random seed to 1 to ensure that anyone rerunning the analysis will obtain the same partition. we then split the dataset into an 80% training set and a 20% hold-out test set, stratifying on the AI-usage flag so that both subsets maintain the same proportion of users and non-users. The larger training portion supports cross-validation and hyperparameter tuning, while the reserved 20% remains untouched until the very end, providing an unbiased estimate of out-of-sample performance.

```r
# Load and prepare data
edx <- read.csv("global_ai_tools_students_use.csv", stringsAsFactors = FALSE) %>%
  mutate(
    uses_ai_for_study = factor(uses_ai_for_study, levels = c("False","True"))
  )
```

3. Exploratory Data Analysis (EDA)

We began by exploring the data to get a clear picture of what we are working with. First, we checked the number of students and the type of each variable. Then we looked at how age, gender, country, and grade are distributed to understand who our students are. After that, we calculated the proportion of students using any AI tool and counted how many use each specific tool to see which ones are most popular. We also examined how often students use multiple tools at once. For each tool, we computed the average usefulness score, its variability, and the percentage of missing ratings (since non-users do not rate tools). We reviewed missing data across all columns to catch any quality issues. Finally, we created plots showing, for example, how adoption varies with age and how tool preference differs by country. This step-by-step exploration helped us uncover important trends and potential challenges before moving on to model building.

3.1 Dimensions and structure of the Dataset

First, we examined the number of rows and columns in the dataset. Then, we checked each variable's type.

```
dim(edx)
```

```
## [1] 500  22
```

```
str(edx)
```

```
## 'data.frame':    500 obs. of  22 variables:
##  $ student_id         : chr  "S0001" "S0002" "S0003" "S0004" ...
##  $ age                : int  17 18 16 18 18 15 16 16 16 18 ...
##  $ gender             : chr  "Female" "Female" "Male" "Female" ...
##  $ country            : chr  "India" "Canada" "UK" "UK" ...
##  $ grade              : chr  "12th" "10th" "12th" "10th" ...
##  $ uses_ai_for_study  : Factor w/ 2 levels "False","True": 2 2 1 1 2 2 2 2 1 2 ...
##  $ uses_chatgpt       : chr  "False" "False" "False" "False" ...
##  $ uses_gemini        : chr  "False" "True" "False" "False" ...
##  $ uses_grammarly     : chr  "False" "False" "False" "False" ...
##  $ uses_quillbot      : chr  "False" "True" "False" "False" ...
##  $ uses_notion_ai     : chr  "True" "False" "False" "False" ...
##  $ uses_phind         : chr  "False" "False" "False" "False" ...
##  $ uses_edu_chat      : chr  "False" "False" "False" "False" ...
##  $ uses_other         : chr  "True" "False" "False" "False" ...
##  $ usefulness_chatgpt  : num  NA NA NA NA NA NA NA NA NA 8 ...
##  $ usefulness_gemini   : num  NA 9 NA NA NA NA NA 7 NA NA ...
##  $ usefulness_grammarly: num  NA NA NA NA 10 NA NA NA NA 6 ...
##  $ usefulness_quillbot : num  NA 10 NA NA NA NA NA 6 NA NA ...
##  $ usefulness_notion_ai: num  6 NA NA NA NA NA NA 10 NA NA ...
##  $ usefulness_phind    : num  NA NA NA NA NA 7 NA 6 NA NA ...
##  $ usefulness_edu_chat : num  NA NA NA NA 8 NA NA NA NA NA ...
##  $ usefulness_other    : num  6 NA NA NA NA NA NA NA NA NA ...
```

3.2 The demographic distributions

We then summarize student demographics by computing the age range and counting how many students fall into each gender, country, and grade category

```
#  Demographic distributions
summary(edx$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   14.00   15.00   16.00   15.97   17.00   18.00
```

```
edx %>% count(gender)
```

```
##       gender   n
## 1     Female 216
## 2       Male 231
## 3 Non-binary  53
```

```
edx %>% count(country)
```

```
##        country  n
## 1    Australia 48
## 2       Brazil 44
## 3       Canada 40
## 4      Germany 48
## 5        India 46
## 6        Japan 51
## 7      Nigeria 56
## 8  South Korea 53
## 9           UK 61
## 10         USA 53
```

```
edx %>% count(grade)
```

```
##   grade   n
## 1  10th 131
## 2  11th 122
## 3  12th 120
## 4   9th 127
```

3.3 Overall AI Adoption Rate

To understand how common AI use is, we tally how many students report using any AI tool and compute the proportion of users versus non-users. We also identified the most and least popular tools at a glance.

```
# Overall AI adoption rate
edx %>%
  count(uses_ai_for_study) %>%
  mutate(prop = n / sum(n))
```

```
##   uses_ai_for_study   n  prop
## 1             False 114 0.228
## 2              True 386 0.772
```

```
# Frequency of each AI tool
edx %>%
  select(starts_with("uses_"), -uses_ai_for_study) %>%
  summarise(across(everything(), ~ sum(. == "True"))) %>%
  pivot_longer(everything(), names_to = "tool", values_to = "count")
```

```
## # A tibble: 8 x 2
##   tool           count
##   <chr>          <int>
```

```
## 1 uses_chatgpt     115
## 2 uses_gemini      123
## 3 uses_grammarly   122
## 4 uses_quillbot    122
## 5 uses_notion_ai   114
## 6 uses_phind       127
## 7 uses_edu_chat    105
## 8 uses_other       100
```

4. Model Development and Evaluation

In this project, we first developed and validated a a Random Forest to predict whether a student uses any AI tool, and evaluate its performance via ROC-AUC and accuracy. Secondly, we build a multiclass classifier among students who adopt AI to predict which specific AI tool(s) they choose, assessing models with overall accuracy and per-class F1-scores.

4.1 Dataset Split for Model Testing

First, we fixed the random seed to 1 so our split would be reproducible. Then we used a stratified sampling approach to allocate 20 percent of the observations to a test cohort which ensuring that the proportion of AI adopters and non-adopters in the test set matched that of the full dataset and retained the remaining 80 percent as our training cohort. Finally, we removed any test records whose gender, country, or grade category did not appear in the training cohort, so that every categorical level in the test set had been seen during model fitting.

```
# Train-test split (80/20 stratified)
set.seed(1)
trainIndex <- createDataPartition(edx$uses_ai_for_study, p = 0.8, list = FALSE)
train      <- edx[trainIndex, ]
test       <- edx[-trainIndex, ]

# Define your tool names & corresponding usefulness columns
tools     <- c("chatgpt", "gemini", "grammarly",
               "quillbot", "notion_ai", "phind",
               "edu_chat", "other")
use_cols  <- paste0("usefulness_", tools)

# Filter to adopters and compute "primary_tool"
df_adopt <- edx %>%
  # keep only rows where uses_ai_for_study == "True"
  filter(uses_ai_for_study == "True") %>%
  # replace NAs with a very low value so they aren't picked
  mutate(across(all_of(use_cols), ~replace_na(.x, -Inf))) %>%
  # for each row, find which usefulness_* is maximal
  rowwise() %>%
  mutate(
    primary_tool = tools[which.max(c_across(all_of(use_cols)))]
  ) %>%
  ungroup() %>%
  # turn into a factor (caret needs this)
  mutate(primary_tool = factor(primary_tool, levels = tools))
```

```r
# Train/test split on adopters (80/20 stratified by primary_tool)
set.seed(1)
idx         <- createDataPartition(df_adopt$primary_tool, p = 0.8, list = FALSE)
train_adopt <- df_adopt[idx, ]
test_adopt  <- df_adopt[-idx, ]
```

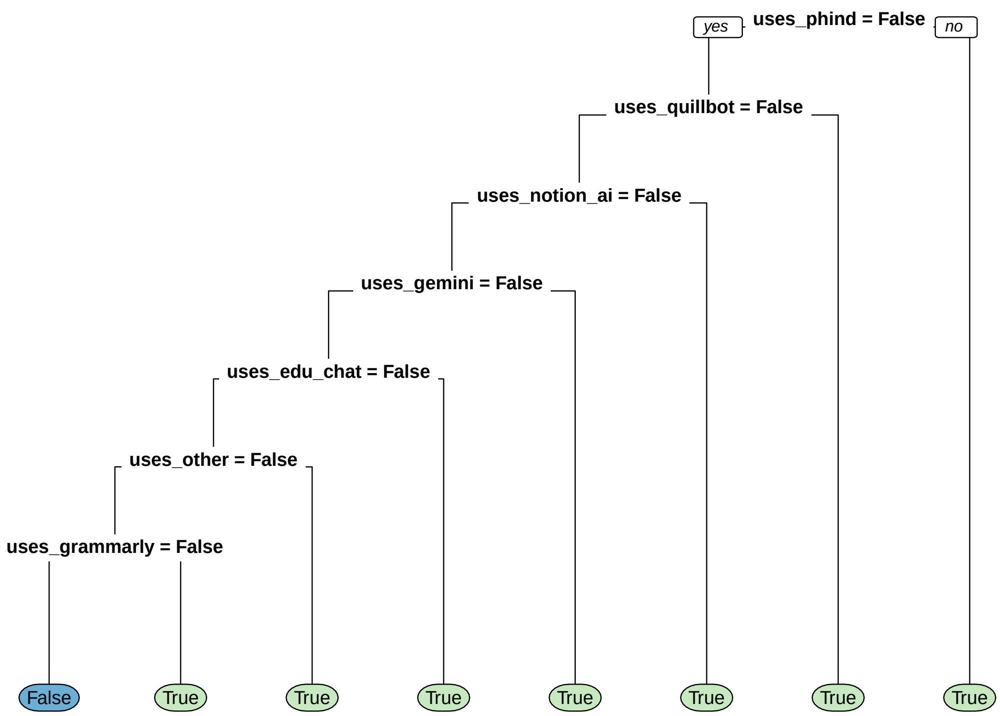
4.2 CART Model for Tool Selection

This section show a code fits a single decision tree model using the CART (Classification and Regression Trees) algorithm via the rpart package. The code builds a classification decision tree to predict whether a student uses AI for study purposes, based on their age, gender, country, grade, and which AI tools they use (e.g., ChatGPT, Grammarly, Notion AI). The tree's growth is controlled to avoid overfitting by using a complexity parameter of 0.01.

```r
# Fit a single CART tree with rpart
fit_tree <- rpart(
  uses_ai_for_study ~ age + gender + country + grade +
    uses_chatgpt + uses_gemini + uses_grammarly +
    uses_quillbot + uses_notion_ai + uses_phind +
    uses_edu_chat + uses_other,
  data    = train,
  method  = "class",
  control = rpart.control(cp = 0.01)
)
```

We then produced a plot of the decision tree to show the essential structure of how the model makes decisions, without cluttering it with details like probabilities or sample counts.

```r
# Visualize the tree
rpart.plot(
  fit_tree,
  type          = 0,    # node labels only (no split text under nodes)
  extra         = 0,    # no class/prob/count info in leaves
  fallen.leaves = TRUE, # align terminal nodes at the same depth
  cex           = 0.6   # smaller text so it never overlaps
)
```



4.2.1 5-fold Cross-Validation with ROC metric

We performed a model tuning and evaluation for a decision tree classifier using the caret package in R. It first sets up a 5-fold cross-validation procedure that evaluates model performance based on the ROC AUC score. A range of values for the complexity parameter (cp) is defined, and the model is trained using these values to find the one that gives the best performance.

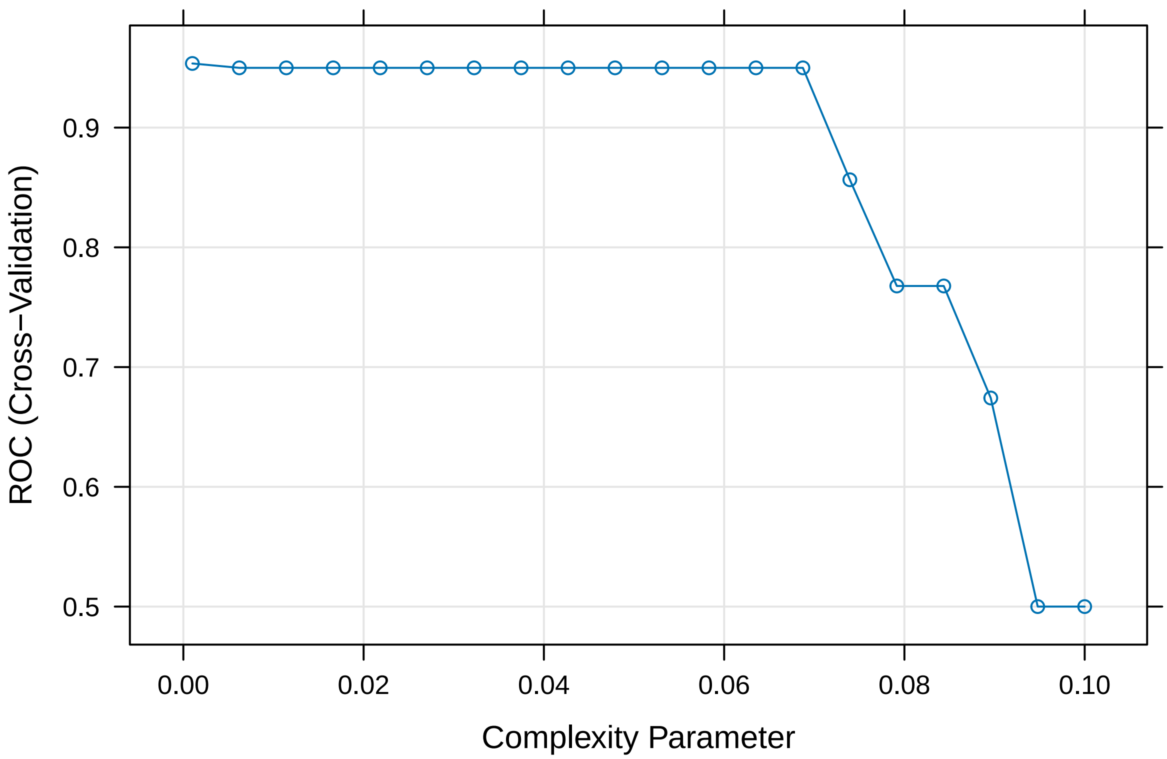
Once the best cp value is identified, the model is retrained using this optimal setting. The final model is then used to make predictions on the test dataset, and its performance is assessed using a confusion matrix.

```
ctrl      <- trainControl(
  method          = "cv",
  number          = 5,
  classProbs      = TRUE,
  summaryFunction = twoClassSummary
)
rpartGrid <- expand.grid(cp = seq(0.001, 0.1, length = 20))

set.seed(1)
train_rpart_cv <- train(
  uses_ai_for_study ~ age + gender + country + grade +
    uses_chatgpt + uses_gemini + uses_grammarly +
    uses_quillbot + uses_notion_ai + uses_phind +
    uses_edu_chat + uses_other,
  data      = train,
```

```r
  method    = "rpart",
  metric    = "ROC",
  trControl = ctrl,
  tuneGrid  = rpartGrid
)

print(train_rpart_cv$bestTune)
```

```
##      cp
## 1 0.001
```

```r
plot(train_rpart_cv)
```



We checked how well the decision tree performs by predicting on unseen data and evaluating the prediction results using standard classification metrics.

```r
pred_tree <- predict(fit_tree, test, type = "class")
print(confusionMatrix(pred_tree, test$uses_ai_for_study))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction False True
##      False    22    4
```

```
##      True      0   73
##
##                Accuracy : 0.9596
##                  95% CI : (0.8998, 0.9889)
##     No Information Rate : 0.7778
##     P-Value [Acc > NIR] : 4.567e-07
##
##                   Kappa : 0.8902
##
##  Mcnemar's Test P-Value : 0.1336
##
##             Sensitivity : 1.0000
##             Specificity : 0.9481
##          Pos Pred Value : 0.8462
##          Neg Pred Value : 1.0000
##              Prevalence : 0.2222
##          Detection Rate : 0.2222
##    Detection Prevalence : 0.2626
##       Balanced Accuracy : 0.9740
##
##        'Positive' Class : False
##
```

The decision tree performs very well, especially for predicting the 'False' class, with 100% sensitivity and 100% NPV. The slight weakness is in specificity and PPV, meaning it sometimes misclassifies True cases as False. The model is highly accurate and reliable, with a Kappa of 0.89 and balanced accuracy of 97.4%, making it a solid choice if simplicity and interpretability are important.

4.2.2 Random Forest Model with Caret Tuning

A Random Forest classification model was developed using the caret package in R to predict whether students use AI for study purposes. The model was tuned using 5-fold cross-validation, where the performance metric was based on the Area Under the ROC Curve (AUC-ROC).

A tuning grid was defined to test four values of the mtry parameter (2, 4, 6, and 8), which controls the number of predictors randomly selected at each tree split. The model was trained using 500 trees to ensure stability in the predictions.

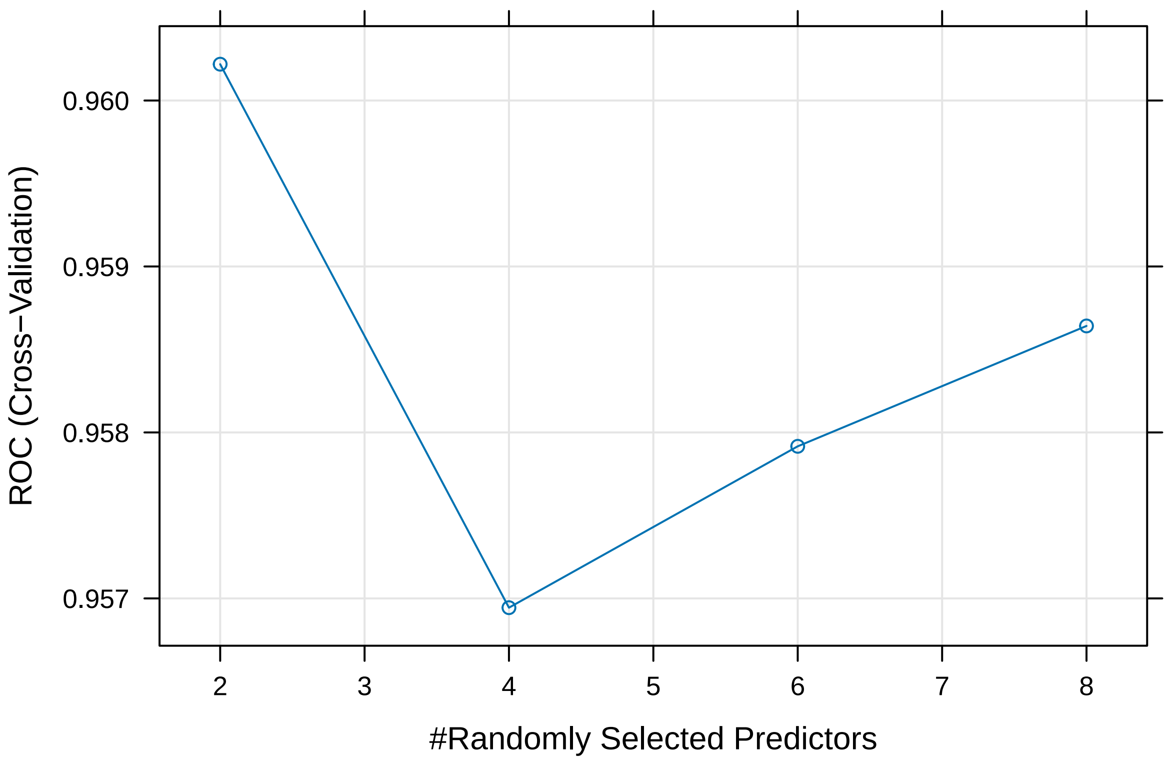
After evaluating performance across the different mtry values, the model with the best ROC score was selected. This optimized model was then used to make predictions on the test dataset. The prediction results were evaluated using a confusion matrix, which provided insights into the model's accuracy, sensitivity, specificity, and other classification metrics.

```r
# Random Forest with caret tuning
rfGrid <- expand.grid(mtry = c(2, 4, 6, 8))

set.seed(1)
train_rf_cv <- train(
  uses_ai_for_study ~ age + gender + country + grade +
    uses_chatgpt + uses_gemini + uses_grammarly +
    uses_quillbot + uses_notion_ai + uses_phind +
    uses_edu_chat + uses_other,
  data       = train,
  method     = "rf",
```

```r
  metric      = "ROC",
  ntree       = 500,
  trControl   = ctrl,
  tuneGrid    = rfGrid,
  importance = TRUE
)

print(train_rf_cv$bestTune)
```

```
##   mtry
## 1    2
```

```r
plot(train_rf_cv)
```




```r
pred_rf_cv <- predict(train_rf_cv, test)
print(confusionMatrix(pred_rf_cv, test$uses_ai_for_study))
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction False True
##      False    20    4
##      True      2   73
##
```

```
##                Accuracy : 0.9394
##                  95% CI : (0.8727, 0.9774)
##     No Information Rate : 0.7778
##     P-Value [Acc > NIR] : 1.213e-05
##
##                   Kappa : 0.8302
##
##  Mcnemar's Test P-Value : 0.6831
##
##             Sensitivity : 0.9091
##             Specificity : 0.9481
##          Pos Pred Value : 0.8333
##          Neg Pred Value : 0.9733
##              Prevalence : 0.2222
##          Detection Rate : 0.2020
##    Detection Prevalence : 0.2424
##       Balanced Accuracy : 0.9286
##
##        'Positive' Class : False
##
```

After tuning the Random Forest model using cross-validation, we tested it on unseen data to check its performance. The model achieved a high accuracy of 93.94%, meaning it correctly predicted most of the students' AI usage behavior.

From the confusion matrix, the model correctly identified 73 students who used AI and 20 who did not. It made only a few mistakes — 2 students who didn't use AI were wrongly predicted as users, and 4 actual users were missed.

The model showed strong results across other metrics too. It had a sensitivity of 90.91%, which means it was good at spotting students who did not use AI. Its specificity was 94.81%, showing it was even better at recognizing students who did use AI. The balanced accuracy was 92.86%, confirming that the model worked well across both groups.

In short, the tuned Random Forest model performed very well and can be confidently used to predict whether students use AI tools for study based on their background and tool usage.

4.3 Predicting Students' Primary AI Tool

In this step, a machine learning model was used to predict which AI tool each student mainly uses. There were eight possible tools to choose from, making this a multiclass prediction task.

The model used information such as the student's age, gender, country, grade level, and which AI tools they have used. It was trained to find patterns in these details to guess the student's main AI tool.

To make sure the model was accurate, it was tested using five rounds of cross-validation. It also tried different settings to find the one that worked best. The final model used 500 decision trees and was also set up to show which features were most important in making the predictions.

This model helps us understand what factors influence students' choice of their primary AI tool.

```r
#Cross-Validation and Tuning Setup

ctrl   <- trainControl(
  method           = "cv",
  number           = 5,
```

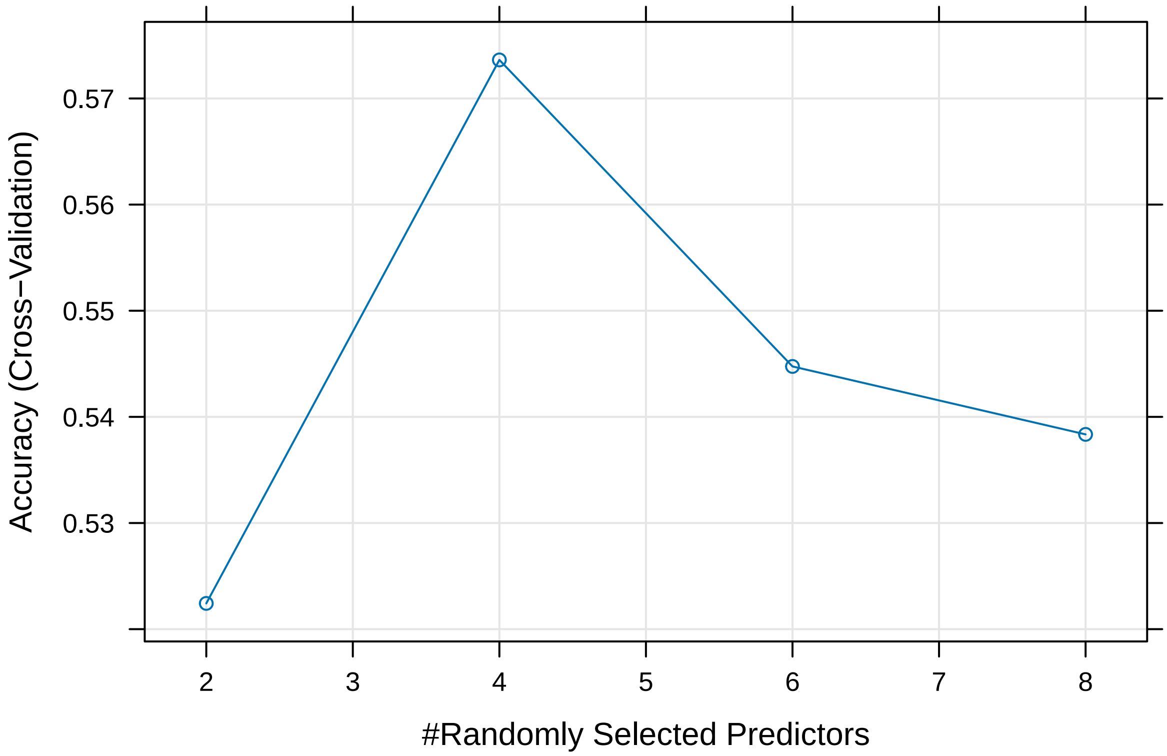
```r
  classProbs      = FALSE,    # multiclass accuracy doesn't need probs
  summaryFunction = defaultSummary
)
rfGrid <- expand.grid(mtry = c(2, 4, 6, 8))

# Fit the multiclass Random Forest

set.seed(1)
rf_multi <- train(
  primary_tool ~ age + gender + country + grade +
    uses_chatgpt + uses_gemini + uses_grammarly +
    uses_quillbot + uses_notion_ai + uses_phind +
    uses_edu_chat + uses_other,
  data      = train_adopt,
  method    = "rf",
  metric    = "Accuracy",
  trControl = ctrl,
  tuneGrid  = rfGrid,
  ntree     = 500,
  importance= TRUE
)
```

After training the model, the results were reviewed to see how well it performed with different settings.

```r
# Inspect tuning results
print(rf_multi)            # shows accuracy by mtry
```

```
## Random Forest
##
## 312 samples
##  12 predictor
##   8 classes: 'chatgpt', 'gemini', 'grammarly', 'quillbot', 'notion_ai', 'phind', 'edu_chat', 'other'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 249, 250, 250, 250, 249
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.5224270  0.4419869
##   4     0.5736303  0.5057798
##   6     0.5447517  0.4728313
##   8     0.5383513  0.4659545
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 4.
```

```r
plot(rf_multi)             # visualizes the tuning curve
```



The model was tested on a new group of students to check its accuracy. It predicted each student's main AI tool, and the results were compared with the actual answers. A confusion matrix showed the overall accuracy and how well the model did for each tool.

```
# Evaluate on the hold-out adopters
pred_multi <- predict(rf_multi, test_adopt)
cm_multi   <- confusionMatrix(pred_multi, test_adopt$primary_tool)
print(cm_multi)            # overall accuracy + per-class stats
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  chatgpt gemini grammarly quillbot notion_ai phind edu_chat other
##   chatgpt        11      0         0        1         0     0        0     0
##   gemini          1      8         2        2         1     2        0     0
##   grammarly       1      0         5        0         0     1        1     1
##   quillbot        0      1         0        6         0     0        1     0
##   notion_ai       1      1         0        0         8     3        1     1
##   phind           0      0         1        1         0     2        1     0
##   edu_chat        1      0         1        0         0     0        3     0
##   other           0      0         0        0         0     0        1     3
##
## Overall Statistics
##
##                Accuracy : 0.6216
##                  95% CI : (0.5013, 0.7319)
```

```
##     No Information Rate : 0.2027
##     P-Value [Acc > NIR] : 5.325e-15
##
##                   Kappa : 0.5629
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: chatgpt Class: gemini Class: grammarly
## Sensitivity                  0.7333        0.8000          0.55556
## Specificity                  0.9831        0.8750          0.93846
## Pos Pred Value               0.9167        0.5000          0.55556
## Neg Pred Value               0.9355        0.9655          0.93846
## Prevalence                   0.2027        0.1351          0.12162
## Detection Rate               0.1486        0.1081          0.06757
## Detection Prevalence         0.1622        0.2162          0.12162
## Balanced Accuracy            0.8582        0.8375          0.74701
##                      Class: quillbot Class: notion_ai Class: phind
## Sensitivity                  0.60000           0.8889      0.25000
## Specificity                  0.96875           0.8923      0.95455
## Pos Pred Value               0.75000           0.5333      0.40000
## Neg Pred Value               0.93939           0.9831      0.91304
## Prevalence                   0.13514           0.1216      0.10811
## Detection Rate               0.08108           0.1081      0.02703
## Detection Prevalence         0.10811           0.2027      0.06757
## Balanced Accuracy            0.78438           0.8906      0.60227
##                      Class: edu_chat Class: other
## Sensitivity                  0.37500      0.60000
## Specificity                  0.96970      0.98551
## Pos Pred Value               0.60000      0.75000
## Neg Pred Value               0.92754      0.97143
## Prevalence                   0.10811      0.06757
## Detection Rate               0.04054      0.04054
## Detection Prevalence         0.06757      0.05405
## Balanced Accuracy            0.67235      0.79275
```

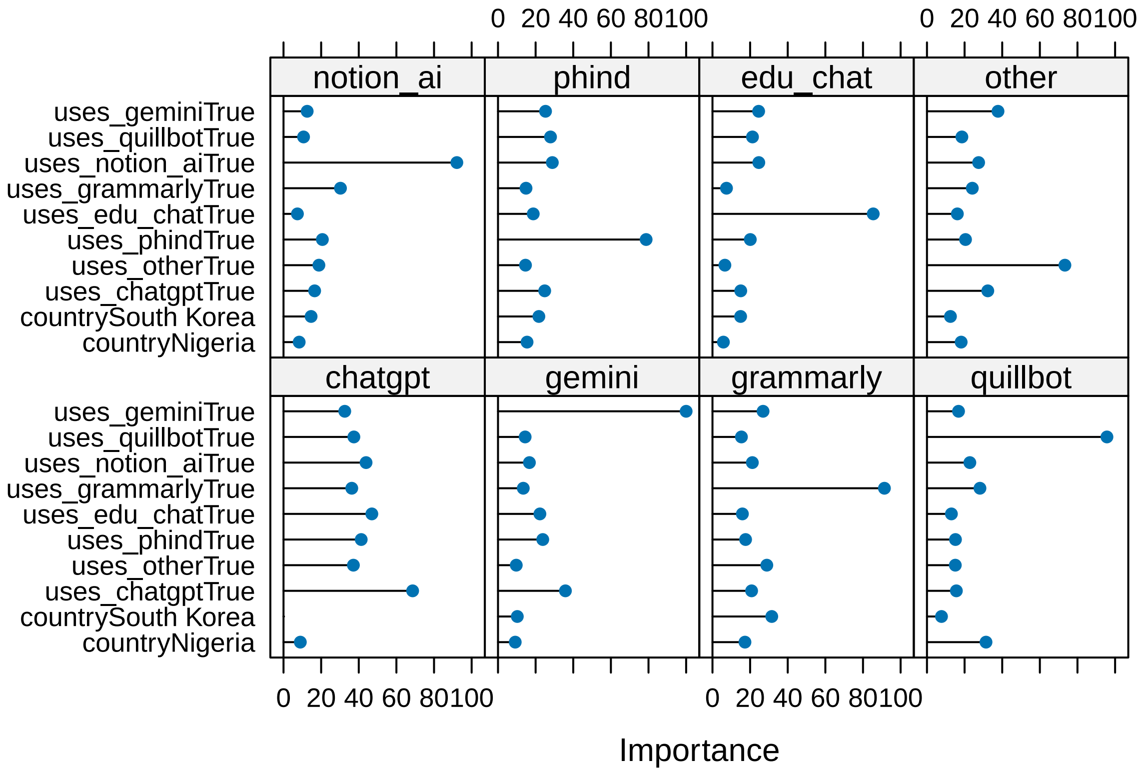
The model was analyzed to see which features were most important in predicting students' main AI tool. A list and graph showed the top 10 factors that influenced the model's decisions the most.

```r
# Variable importance
vi_multi <- varImp(rf_multi)
print(vi_multi)            # which features drive tool choice
```

```
## rf variable importance
##
##   variables are sorted by maximum importance across the classes
##   only 20 most important variables shown (out of 23)
##
##                     chatgpt  gemini grammarly quillbot notion_ai  phind edu_chat
## uses_geminiTrue      32.554 100.000    26.894   16.779    12.577 25.268   24.538
## uses_quillbotTrue    37.406  14.429    15.385   95.667    10.652 27.923   21.261
## uses_notion_aiTrue   43.871  16.695    21.198   22.804    92.137 28.887   24.619
```

```
## uses_grammarlyTrue  36.285  13.445    91.393   28.167    30.281 14.892    7.446
## uses_edu_chatTrue   46.951  22.280    15.904   13.002     7.400 18.731   85.450
## uses_phindTrue      41.292  23.797    17.602   15.153    20.658 78.737   20.103
## uses_otherTrue      37.149   9.714    28.886   15.063    18.813 14.614    6.603
## uses_chatgptTrue    68.630  35.830    20.798   15.629    16.546 24.825   15.037
## countrySouth Korea   0.000  10.269    31.506    7.725    14.671 21.728   14.955
## countryNigeria       8.942   9.161    17.268   31.389     8.327 15.455    5.802
## grade9th            13.683  23.213    13.690   18.062    19.827 15.296   31.044
## grade12th           14.578  18.876    15.200   14.514    24.660  7.528   14.588
## age                 13.092  10.249    13.652   14.001    19.185 23.570   11.518
## countryUSA           7.762  13.954    20.499   11.701    22.591  5.346    4.790
## countryJapan        15.568  19.438    11.971   22.831     9.842 17.558    8.143
## genderMale          11.172   9.945     7.749   18.996    14.862  9.487   22.188
## countryBrazil       10.135  10.975    19.351   22.184    10.527  7.749   17.538
## countryIndia         9.383  21.932    11.770   14.977    12.722 15.409   18.943
## countryCanada       16.141  16.426     8.658   11.675    13.556  6.313   11.538
## countryUK           15.006  17.633    10.246   15.466    11.878 19.934    8.965
##                     other
## uses_geminiTrue    37.758
## uses_quillbotTrue  18.564
## uses_notion_aiTrue 27.464
## uses_grammarlyTrue 24.104
## uses_edu_chatTrue  16.161
## uses_phindTrue     20.493
## uses_otherTrue     73.315
## uses_chatgptTrue   32.331
## countrySouth Korea 12.487
## countryNigeria     18.188
## grade9th            9.330
## grade12th           1.436
## age                10.437
## countryUSA         23.329
## countryJapan       14.866
## genderMale         16.013
## countryBrazil      12.788
## countryIndia       12.832
## countryCanada      21.287
## countryUK          10.160
```

```r
plot(vi_multi, top = 10)   # plot the top 10 most important
```



5 Conclusion

Our first model was able to tell who uses AI almost perfectly, and our second model correctly guessed each user's favorite tool most of the time. In other words, just knowing a student's basic demographics and which AI services they already use lets us predict both whether they will adopt AI and which tool they will choose. These findings could help schools offer personalized recommendations or support to students as they explore different AI resources. For the further studies, I recommend to construct a regression model to predict the student's self-reported usefulness score and evaluate the predictive accuracy using RMSE and R square.

Reference

Daksh Bhatnagar. (2025). AI Tools Usage Among Global High School Students [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DS/7656698