```r
# This R environment comes with many helpful analytics packages
installed
# It is defined by the kaggle/rstats Docker image:
https://github.com/kaggle/docker-rstats
# For example, here's a helpful package to load

library(tidyverse) # metapackage of all tidyverse packages

# Input data files are available in the read-only "../input/"
directory
# For example, running this (by clicking run or pressing Shift+Enter)
will list all files under the input directory

list.files(path = "../input")

# You can write up to 20GB to the current directory (/kaggle/working/)
that gets preserved as output when you create a version using "Save &
Run All"
# You can also write temporary files to /kaggle/temp/, but they won't
be saved outside of the current session
```

```
── Attaching core tidyverse packages ──────────────────────
tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.4
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.4.4     ✔ tibble    3.2.1
✔ lubridate 1.9.3     ✔ tidyr     1.3.0
✔ purrr     1.0.2
── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to
force all conflicts to become errors
```

```
[1] "studentpred"
```

1) Motivation: For my final project, I would like to focus on whether studeents are likely to graduate from college or dropout based on a number of predictors. I believe this task is important because it will aid professors/institutions in making timely decisions and changes to improve a student's success rate. Additionally, my techique **will** outperform a human's judgement on a student's potential success. A professor or an institution will not know everything going on in a student's life that may affect whether they dropout of school or not, let alone thousand's of students. Relying solely on humans is a disservice to the students because without these predictions, students who may potentially dropout are not prioritized when they should be getting all the extra help they can. Therefore, I believe my techique will help making these predictions easier overall, thus, bettering student success rates as well.

2) Exploratory Data Analysis: In this section, I will explore, define, and clean the data if needed.

2.1. Data Cleaning: I will clean up any data if necessary.

```r
# loading in data
data <- read_csv('../input/studentpred/data.csv')
```

```
Rows: 4424 Columns: 1
── Column specification
────────────────────────────────────────────────────────
Delimiter: ","
chr (1): Marital status;Application mode;Application
order;Course;"Daytime/e...

ⓘ Use `spec()` to retrieve the full column specification for this
data.
ⓘ Specify the column types or set `show_col_types = FALSE` to quiet
this message.
```

```r
# viewing data
dim(data)
glimpse(data)
head(data)
summary(data)
```

```
[1] 4424    1
```

```
Rows: 4,424
Columns: 1
$ `Marital status;Application mode;Application
order;Course;"Daytime/evening attendance\t";Previous
qualification;Previous qualification (grade);Nacionality;Mother's
qualification;Father's qualification;Mother's occupation;Father's
occupation;Admission grade;Displaced;Educational special
needs;Debtor;Tuition fees up to date;Gender;Scholarship holder;Age at
enrollment;International;Curricular units 1st sem
(credited);Curricular units 1st sem (enrolled);Curricular units 1st
sem (evaluations);Curricular units 1st sem (approved);Curricular units
1st sem (grade);Curricular units 1st sem (without
evaluations);Curricular units 2nd sem (credited);Curricular units 2nd
sem (enrolled);Curricular units 2nd sem (evaluations);Curricular units
2nd sem (approved);Curricular units 2nd sem (grade);Curricular units
2nd sem (without evaluations);Unemployment rate;Inflation
rate;GDP;Target` <chr> …

  Marital status;Application mode;Application
order;Course;"Daytime/evening attendance\t";Previous
qualification;Previous qualification (grade);Nacionality;Mother's
qualification;Father's qualification;Mother's occupation;Father's
occupation;Admission grade;Displaced;Educational special
needs;Debtor;Tuition fees up to date;Gender;Scholarship holder;Age at
enrollment;International;Curricular units 1st sem
(credited);Curricular units 1st sem (enrolled);Curricular units 1st
sem (evaluations);Curricular units 1st sem (approved);Curricular units
```

```
1st sem (grade);Curricular units 1st sem (without
evaluations);Curricular units 2nd sem (credited);Curricular units 2nd
sem (enrolled);Curricular units 2nd sem (evaluations);Curricular units
2nd sem (approved);Curricular units 2nd sem (grade);Curricular units
2nd sem (without evaluations);Unemployment rate;Inflation
rate;GDP;Target
1
1;17;5;171;1;1;122.0;1;19;12;5;9;127.3;1;0;0;1;1;0;20;0;0;0;0;0;0.0;0;
0;0;0;0;0.0;0;10.8;1.4;1.74;Dropout

2
1;15;1;9254;1;1;160.0;1;1;3;3;3;142.5;1;0;0;0;1;0;19;0;0;6;6;6;14.0;0;
0;6;6;6;13.666666666666666;0;13.9;-0.3;0.79;Graduate

3
1;1;5;9070;1;1;122.0;1;37;37;9;9;124.8;1;0;0;0;1;0;19;0;0;6;0;0;0.0;0;
0;6;0;0;0.0;0;10.8;1.4;1.74;Dropout

4
1;17;2;9773;1;1;122.0;1;38;37;5;3;119.6;1;0;0;1;0;0;20;0;0;6;8;6;13.42
8571428571429;0;0;6;10;5;12.4;0;9.4;-0.8;-3.12;Graduate

5
2;39;1;8014;0;1;100.0;1;37;38;9;9;141.5;0;0;0;1;0;0;45;0;0;6;9;5;12.33
3333333333334;0;0;6;6;6;13.0;0;13.9;-0.3;0.79;Graduate

6
2;39;1;9991;0;19;133.1;1;37;37;9;7;114.8;0;0;1;1;1;0;50;0;0;5;10;5;11.
857142857142858;0;0;5;17;5;11.5;5;16.2;0.3;-0.92;Graduate

 Marital status;Application mode;Application
order;Course;"Daytime/evening attendance\t";Previous
qualification;Previous qualification (grade);Nacionality;Mother's
qualification;Father's qualification;Mother's occupation;Father's
occupation;Admission grade;Displaced;Educational special
needs;Debtor;Tuition fees up to date;Gender;Scholarship holder;Age at
enrollment;International;Curricular units 1st sem
(credited);Curricular units 1st sem (enrolled);Curricular units 1st
sem (evaluations);Curricular units 1st sem (approved);Curricular units
1st sem (grade);Curricular units 1st sem (without
evaluations);Curricular units 2nd sem (credited);Curricular units 2nd
sem (enrolled);Curricular units 2nd sem (evaluations);Curricular units
2nd sem (approved);Curricular units 2nd sem (grade);Curricular units
2nd sem (without evaluations);Unemployment rate;Inflation
rate;GDP;Target
 Length:4424

 Class :character

 Mode  :character
```

When viewing my data like usually, I noticed that this dataset only has one column, so I have to manually divide the columns myself.

```
# giving the single column a name
colnames(data) <- c("x")

# separating the columns
data <- data %>%
separate_wider_delim(x, delim = ";", names = c("marital_status",
"app_mode", "app_order",
                                               "course",
"attendance_time", "prev_qualification",

"prev_qualification_grade", "nationality", "mot_qualification",
                                               "fat_qualification",
"mot_occupation", "fat_occupation",
                                                "admission_grade",
"displaced", "special_needs", "debtor",

"tuition_fees_up_to_date", "gender", "scholarship_holder",
                                               "age_at_enrollment",
"international", "1st_sem_credited",
                                               "1st_sem_enrolled",
"1st_sem_eval", "1st_sem_approved",
                                               "1st_sem_grade",
"1st_sem_no_eval", "2nd_sem_credited",
                                               "2nd_sem_enrolled",
"2nd_sem_eval", "2nd_sem_approved",
                                               "2nd_sem_grade",
"2nd_sem_no_eval", "unemployment_rate",
                                               "inflation_rate", "GDP",
"target"))

# viewing data again
dim(data)
glimpse(data)
head(data)
summary(data)
```

```
[1] 4424   37
```

```
Rows: 4,424
Columns: 37
$ marital_status           <chr> "1", "1", "1", "1", "2", "2", "1",
"1", "1", …
$ app_mode                 <chr> "17", "15", "1", "17", "39", "39",
"1", "18",…
$ app_order                <chr> "5", "1", "5", "2", "1", "1", "1",
"4", "3", …
$ course                   <chr> "171", "9254", "9070", "9773",
```

```
"8014", "9991"…
$ attendance_time          <chr> "1", "1", "1", "1", "0", "0", "1",
"1", "1", …
$ prev_qualification       <chr> "1", "1", "1", "1", "1", "19", "1",
"1", "1",…
$ prev_qualification_grade <chr> "122.0", "160.0", "122.0", "122.0",
"100.0", …
$ nationality              <chr> "1", "1", "1", "1", "1", "1", "1",
"1", "62",…
$ mot_qualification        <chr> "19", "1", "37", "38", "37", "37",
"19", "37"…
$ fat_qualification        <chr> "12", "3", "37", "37", "38", "37",
"38", "37"…
$ mot_occupation           <chr> "5", "3", "9", "5", "9", "9", "7",
"9", "9", …
$ fat_occupation           <chr> "9", "3", "9", "3", "9", "7", "10",
"9", "9",…
$ admission_grade          <chr> "127.3", "142.5", "124.8", "119.6",
"141.5", …
$ displaced                <chr> "1", "1", "1", "1", "0", "0", "1",
"1", "0", …
$ special_needs            <chr> "0", "0", "0", "0", "0", "0", "0",
"0", "0", …
$ debtor                   <chr> "0", "0", "0", "0", "0", "1", "0",
"0", "0", …
$ tuition_fees_up_to_date  <chr> "1", "0", "0", "1", "1", "1", "1",
"0", "1", …
$ gender                   <chr> "1", "1", "1", "0", "0", "1", "0",
"1", "0", …
$ scholarship_holder       <chr> "0", "0", "0", "0", "0", "0", "1",
"0", "1", …
$ age_at_enrollment        <chr> "20", "19", "19", "20", "45", "50",
"18", "22…
$ international            <chr> "0", "0", "0", "0", "0", "0", "0",
"0", "1", …
$ `1st_sem_credited`       <chr> "0", "0", "0", "0", "0", "0", "0",
"0", "0", …
$ `1st_sem_enrolled`       <chr> "0", "6", "6", "6", "6", "5", "7",
"5", "6", …
$ `1st_sem_eval`           <chr> "0", "6", "0", "8", "9", "10", "9",
"5", "8",…
$ `1st_sem_approved`       <chr> "0", "6", "0", "6", "5", "5", "7",
"0", "6", …
$ `1st_sem_grade`          <chr> "0.0", "14.0", "0.0",
"13.428571428571429", "…
$ `1st_sem_no_eval`        <chr> "0", "0", "0", "0", "0", "0", "0",
"0", "0", …
$ `2nd_sem_credited`       <chr> "0", "0", "0", "0", "0", "0", "0",
"0", "0", …
```

```
$ `2nd_sem_enrolled`       <chr> "0", "6", "6", "6", "6", "5", "8",
"5", "6", …
$ `2nd_sem_eval`           <chr> "0", "6", "0", "10", "6", "17", "8",
"5", "7"…
$ `2nd_sem_approved`       <chr> "0", "6", "0", "5", "6", "5", "8",
"0", "6", …
$ `2nd_sem_grade`          <chr> "0.0", "13.666666666666666", "0.0",
"12.4", "…
$ `2nd_sem_no_eval`        <chr> "0", "0", "0", "0", "0", "5", "0",
"0", "0", …
$ unemployment_rate        <chr> "10.8", "13.9", "10.8", "9.4",
"13.9", "16.2"…
$ inflation_rate           <chr> "1.4", "-0.3", "1.4", "-0.8", "-0.3",
"0.3", …
$ GDP                      <chr> "1.74", "0.79", "1.74", "-3.12",
"0.79", "-0.…
$ target                   <chr> "Dropout", "Graduate", "Dropout",
"Graduate",…

  marital_status app_mode app_order course attendance_time
prev_qualification
1 1              17       5         171    1               1

2 1              15       1         9254   1               1

3 1              1        5         9070   1               1

4 1              17       2         9773   1               1

5 2              39       1         8014   0               1

6 2              39       1         9991   0               19

  prev_qualification_grade nationality mot_qualification
fat_qualification …
1 122.0                     1           19                12
…
2 160.0                     1           1                 3
…
3 122.0                     1           37                37
…
4 122.0                     1           38                37
…
5 100.0                     1           37                38
…
6 133.1                     1           37                37
…
  2nd_sem_credited 2nd_sem_enrolled 2nd_sem_eval 2nd_sem_approved
1 0                0                0            0
2 0                6                6            6
```

```
3 0                 6                 0            0
4 0                 6                10            5
5 0                 6                 6            6
6 0                 5                17            5
  2nd_sem_grade      2nd_sem_no_eval unemployment_rate inflation_rate
GDP
1 0.0                 0                10.8              1.4
1.74
2 13.666666666666666 0                13.9              -0.3
0.79
3 0.0                 0                10.8              1.4
1.74
4 12.4                0                9.4               -0.8
-3.12
5 13.0                0                13.9              -0.3
0.79
6 11.5                5                16.2              0.3
-0.92
  target
1 Dropout
2 Graduate
3 Dropout
4 Graduate
5 Graduate
6 Graduate

 marital_status       app_mode          app_order            course

 Length:4424        Length:4424        Length:4424        Length:4424

 Class :character   Class :character   Class :character
Class :character
 Mode  :character   Mode  :character   Mode  :character
Mode  :character
 attendance_time    prev_qualification prev_qualification_grade
 Length:4424        Length:4424        Length:4424
 Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character
 nationality        mot_qualification  fat_qualification
mot_occupation
 Length:4424        Length:4424        Length:4424        Length:4424

 Class :character   Class :character   Class :character
Class :character
 Mode  :character   Mode  :character   Mode  :character
Mode  :character
 fat_occupation     admission_grade     displaced
special_needs
 Length:4424        Length:4424        Length:4424        Length:4424
```

```
 Class :character   Class :character   Class :character
Class :character
 Mode  :character   Mode  :character   Mode  :character
Mode  :character
    debtor          tuition_fees_up_to_date    gender
 Length:4424        Length:4424             Length:4424
 Class :character   Class :character        Class :character
 Mode  :character   Mode  :character        Mode  :character
 scholarship_holder age_at_enrollment  international
1st_sem_credited
 Length:4424        Length:4424        Length:4424        Length:4424

 Class :character   Class :character   Class :character
Class :character
 Mode  :character   Mode  :character   Mode  :character
Mode  :character
 1st_sem_enrolled   1st_sem_eval       1st_sem_approved
1st_sem_grade
 Length:4424        Length:4424        Length:4424        Length:4424

 Class :character   Class :character   Class :character
Class :character
 Mode  :character   Mode  :character   Mode  :character
Mode  :character
 1st_sem_no_eval    2nd_sem_credited   2nd_sem_enrolled   2nd_sem_eval

 Length:4424        Length:4424        Length:4424        Length:4424

 Class :character   Class :character   Class :character
Class :character
 Mode  :character   Mode  :character   Mode  :character
Mode  :character
 2nd_sem_approved   2nd_sem_grade      2nd_sem_no_eval
unemployment_rate
 Length:4424        Length:4424        Length:4424        Length:4424

 Class :character   Class :character   Class :character
Class :character
 Mode  :character   Mode  :character   Mode  :character
Mode  :character
 inflation_rate         GDP               target
 Length:4424        Length:4424        Length:4424
 Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character
```

All of my columns are divided now. Next, I am going to make a correlation plot because I noticed a lot of the variables are very similar to one another.

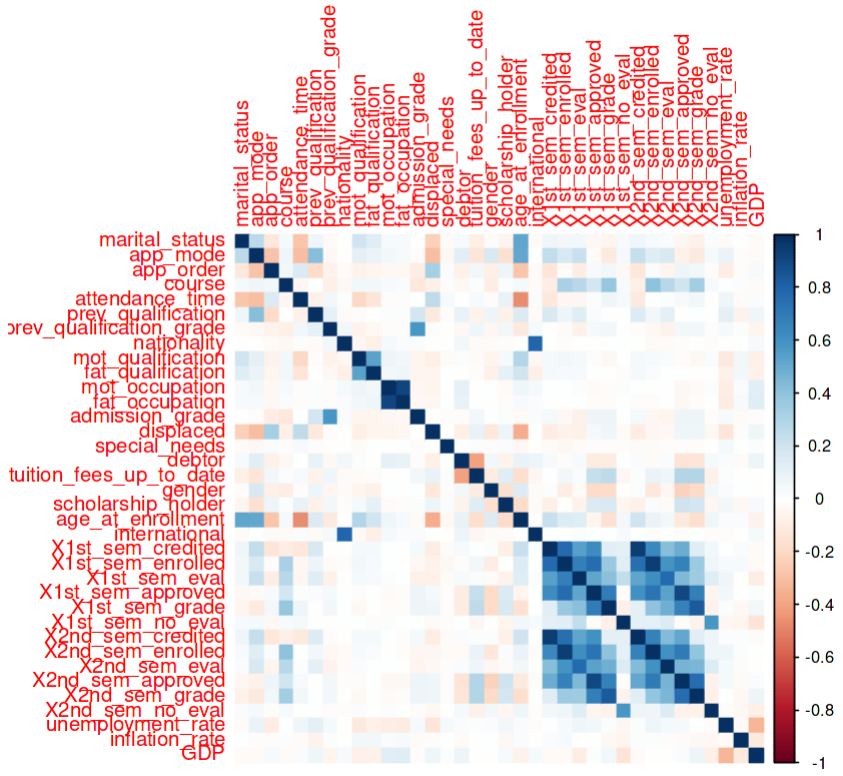
```r
# making variables numeric for correlation plot
fCor <- data %>%
       select(-target)

fCor <- as.data.frame(lapply(fCor, as.numeric))

# viewing
glimpse(fCor)
```

```
Rows: 4,424
Columns: 36
$ marital_status           <dbl> 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1,
1, 1, 1, …
$ app_mode                 <dbl> 17, 15, 1, 17, 39, 39, 1, 18, 1, 1,
1, 1, 1, …
$ app_order                <dbl> 5, 1, 5, 2, 1, 1, 1, 4, 3, 1, 1, 1,
2, 1, 1, …
$ course                   <dbl> 171, 9254, 9070, 9773, 8014, 9991,
9500, 9254…
$ attendance_time          <dbl> 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,
1, 1, 1, …
$ prev_qualification       <dbl> 1, 1, 1, 1, 1, 19, 1, 1, 1, 1, 1, 1,
1, 42, 1…
$ prev_qualification_grade <dbl> 122.0, 160.0, 122.0, 122.0, 100.0,
133.1, 142…
$ nationality              <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 62, 1, 1, 1,
1, 1, 1,…
$ mot_qualification        <dbl> 19, 1, 37, 38, 37, 37, 19, 37, 1, 1,
38, 19, …
$ fat_qualification        <dbl> 12, 3, 37, 37, 38, 37, 38, 37, 1, 19,
19, 38,…
$ mot_occupation           <dbl> 5, 3, 9, 5, 9, 9, 7, 9, 9, 4, 5, 9,
4, 4, 5, …
$ fat_occupation           <dbl> 9, 3, 9, 3, 9, 7, 10, 9, 9, 7, 7, 9,
9, 7, 5,…
$ admission_grade          <dbl> 127.3, 142.5, 124.8, 119.6, 141.5,
114.8, 128…
$ displaced                <dbl> 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1,
1, 1, 1, …
$ special_needs            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, …
$ debtor                   <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,
0, 0, 0, …
$ tuition_fees_up_to_date  <dbl> 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1,
1, 1, 1, …
$ gender                   <dbl> 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
0, 0, 0, …
$ scholarship_holder       <dbl> 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1,
0, 1, 1, …
$ age_at_enrollment        <dbl> 20, 19, 19, 20, 45, 50, 18, 22, 21,
```

```
18, 18, 1…
$ international             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, …
$ X1st_sem_credited         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, …
$ X1st_sem_enrolled         <dbl> 0, 6, 6, 6, 6, 5, 7, 5, 6, 6, 6, 8,
6, 6, 5, …
$ X1st_sem_eval             <dbl> 0, 6, 0, 8, 9, 10, 9, 5, 8, 9, 6, 8,
6, 7, 7,…
$ X1st_sem_approved         <dbl> 0, 6, 0, 6, 5, 5, 7, 0, 6, 5, 6, 7,
0, 6, 4, …
$ X1st_sem_grade            <dbl> 0.00000, 14.00000, 0.00000, 13.42857,
12.3333…
$ X1st_sem_no_eval          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, …
$ X2nd_sem_credited         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, …
$ X2nd_sem_enrolled         <dbl> 0, 6, 6, 6, 6, 5, 8, 5, 6, 6, 6, 8,
6, 6, 5, …
$ X2nd_sem_eval             <dbl> 0, 6, 0, 10, 6, 17, 8, 5, 7, 14, 7,
8, 0, 8, …
$ X2nd_sem_approved         <dbl> 0, 6, 0, 5, 6, 5, 8, 0, 6, 2, 5, 7,
0, 5, 5, …
$ X2nd_sem_grade            <dbl> 0.00000, 13.66667, 0.00000, 12.40000,
13.0000…
$ X2nd_sem_no_eval          <dbl> 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0,
0, 0, 0, …
$ unemployment_rate         <dbl> 10.8, 13.9, 10.8, 9.4, 13.9, 16.2,
15.5, 15.5…
$ inflation_rate            <dbl> 1.4, -0.3, 1.4, -0.8, -0.3, 0.3, 2.8,
2.8, 0.…
$ GDP                       <dbl> 1.74, 0.79, 1.74, -3.12, 0.79, -0.92,
-4.06, …
```

```r
library(corrplot)

correlation_matrix <- cor(fCor)

#print(correlation_matrix)

corrplot(correlation_matrix, method = "color")
```

```
corrplot 0.92 loaded
```

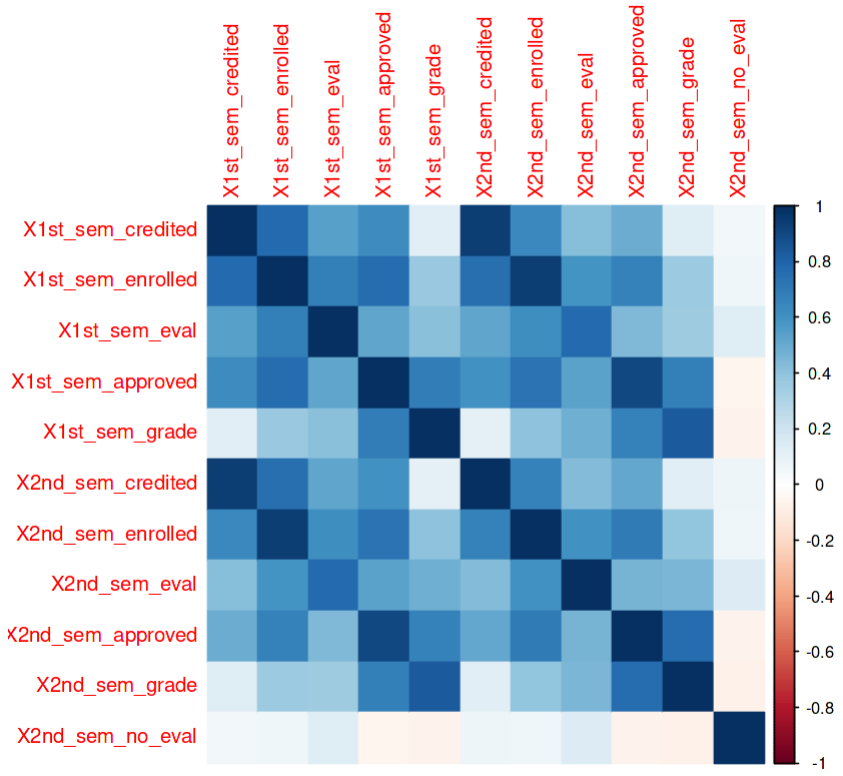


Looks like there is a heavy correlation between mother and father qualifications/occupations, as well as 1st sem credited to 2nd_sem_no_eval so I will make two different plots.

```
# 1st sem credited to 2nd sem no evaluation
selected_columns <- fCor[, c("X1st_sem_credited", "X1st_sem_enrolled",
                             "X1st_sem_eval", "X1st_sem_approved",
"X1st_sem_grade",
                             "X2nd_sem_credited", "X2nd_sem_enrolled",
"X2nd_sem_eval",
                              "X2nd_sem_approved", "X2nd_sem_grade",
"X2nd_sem_no_eval")]

correlation_matrix <- cor(selected_columns)

print(correlation_matrix)

corrplot(correlation_matrix, method = "color")
```

```
                   X1st_sem_credited X1st_sem_enrolled X1st_sem_eval
X1st_sem_credited        1.00000000        0.77434419     0.5429194
X1st_sem_enrolled        0.77434419        1.00000000     0.6802196
X1st_sem_eval            0.54291944        0.68021964     1.0000000
X1st_sem_approved        0.62839442        0.76908348     0.5223961
X1st_sem_grade           0.12297757        0.37699588     0.4180382
X2nd_sem_credited        0.94481104        0.75374671     0.5221865
```

```
X2nd_sem_enrolled        0.64482590        0.94262669     0.6118417
X2nd_sem_eval            0.42784508        0.59956670     0.7788631
X2nd_sem_approved        0.49047789        0.67334052     0.4422653
X2nd_sem_grade           0.13297056        0.36195891     0.3550359
X2nd_sem_no_eval         0.05525634        0.06954735     0.1342959
                  X1st_sem_approved X1st_sem_grade X2nd_sem_credited
X1st_sem_credited         0.6283944     0.12297757        0.94481104
X1st_sem_enrolled         0.7690835     0.37699588        0.75374671
X1st_sem_eval             0.5223961     0.41803818        0.52218653
X1st_sem_approved         1.0000000     0.69611327        0.60766119
X1st_sem_grade            0.6961133     1.00000000        0.11393731
X2nd_sem_credited         0.6076612     0.11393731        1.00000000
X2nd_sem_enrolled         0.7337719     0.40616667        0.67625783
X2nd_sem_eval             0.5399343     0.48723561        0.43097776
X2nd_sem_approved         0.9040021     0.67333493        0.51908105
X2nd_sem_grade            0.6855602     0.83716974        0.12976992
X2nd_sem_no_eval         -0.0539830    -0.06607618        0.07014834
                  X2nd_sem_enrolled X2nd_sem_eval X2nd_sem_approved
X1st_sem_credited        0.64482590     0.4278451        0.49047789
X1st_sem_enrolled        0.94262669     0.5995667        0.67334052
X1st_sem_eval            0.61184174     0.7788631        0.44226525
X1st_sem_approved        0.73377188     0.5399343        0.90400210
X1st_sem_grade           0.40616667     0.4872356        0.67333493
X2nd_sem_credited        0.67625783     0.4309778        0.51908105
X2nd_sem_enrolled        1.00000000     0.6048211        0.70325807
X2nd_sem_eval            0.60482108     1.0000000        0.46353548
X2nd_sem_approved        0.70325807     0.4635355        1.00000000
X2nd_sem_grade           0.39513489     0.4533940        0.76080418
X2nd_sem_no_eval         0.06769748     0.1448774       -0.06156671
                  X2nd_sem_grade X2nd_sem_no_eval
X1st_sem_credited     0.13297056       0.05525634
X1st_sem_enrolled     0.36195891       0.06954735
X1st_sem_eval         0.35503589       0.13429591
X1st_sem_approved     0.68556019      -0.05398300
X1st_sem_grade        0.83716974      -0.06607618
X2nd_sem_credited     0.12976992       0.07014834
X2nd_sem_enrolled     0.39513489       0.06769748
X2nd_sem_eval         0.45339403       0.14487740
X2nd_sem_approved     0.76080418      -0.06156671
X2nd_sem_grade        1.00000000      -0.07921597
X2nd_sem_no_eval     -0.07921597       1.00000000
```



I found a number of correlations over 0.9, so I will get rid of one of the variables where the correlation is too high:

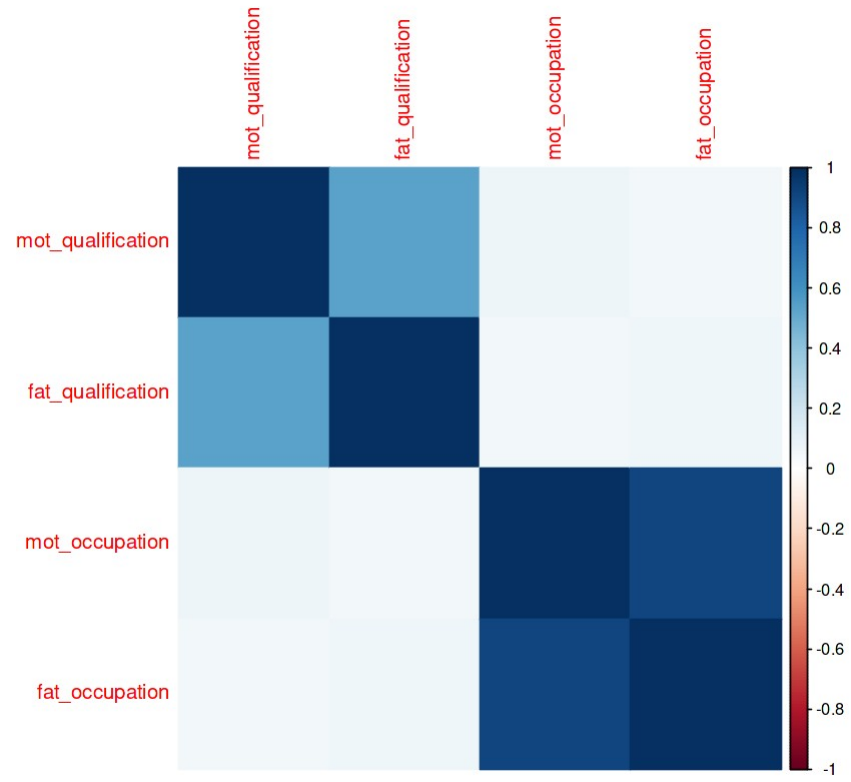
- X1st_sem_credited and X2nd_sem_credited
- X1st_sem_enrolled and X2nd_sem_enrolled
- X1st_sem_approved and X2nd_sem_approved

I will likely get rid of the **2nd semester** variables.

```
selected_columns <- fCor[, c("mot_qualification", "fat_qualification",
"mot_occupation",
                             "fat_occupation")]

correlation_matrix <- cor(selected_columns)

print(correlation_matrix)

corrplot(correlation_matrix, method = "color")
```

```
                  mot_qualification fat_qualification mot_occupation
mot_qualification        1.00000000        0.53513968     0.07677173
fat_qualification        0.53513968        1.00000000     0.05491134
mot_occupation           0.07677173        0.05491134     1.00000000
fat_occupation           0.05232861        0.06304346     0.91047211
                  fat_occupation
```

```
mot_qualification     0.05232861
fat_qualification     0.06304346
mot_occupation        0.91047211
fat_occupation        1.00000000
```



I found that mot_occupation and fat_occupation are correlated over 0.9, so I will get rid of one. I will get rid of the **father occupation** to avoid multicollinearity.

```
# removing the variables
data <- data %>%
select(-c('fat_occupation', '2nd_sem_credited', '2nd_sem_enrolled',
'2nd_sem_approved'))

#head(data)
```

Since I removed the variables that could have caused multicollinearity, I will proceed to make categorical variables factors and continuous numbers numeric. I will leave target as a character variable for now for visualization.

```
# making some variables factors and others numeric -- leaving target
as a chr for now
data <- data %>%
mutate(across(-c('prev_qualification_grade', 'admission_grade',
```

```
'age_at_enrollment',
                 '1st_sem_grade', '2nd_sem_grade',
'unemployment_rate', 'inflation_rate',
                 'GDP', 'target'), as.factor)) %>%
mutate_at(vars('prev_qualification_grade', 'admission_grade',
'age_at_enrollment',
               '1st_sem_grade', '2nd_sem_grade', 'unemployment_rate',
'inflation_rate',
               'GDP'), as.numeric)

#head(data)
```

I will also create a function to rename some of the columns and to move the outcome variable (target) to be the last column in the data set.

```
# creating a function for these modifications

mod <- function(df) {

    # renaming to avoid error
     df <- df %>%
        rename("fst_sem_credited" = "1st_sem_credited",
               "fst_sem_enrolled" = "1st_sem_enrolled",
               "fst_sem_eval" = "1st_sem_eval",
              "fst_sem_approved" = "1st_sem_approved",
              "fst_sem_grade" = "1st_sem_grade",
              "fst_sem_no_eval" = "1st_sem_no_eval",
              "snd_sem_eval" = "2nd_sem_eval",
              "snd_sem_grade" = "2nd_sem_grade",
              "snd_sem_no_eval" = "2nd_sem_no_eval")

    # moved the outcome variable to the end
    df <- df %>%
    relocate(target, .after = last_col())

    return(df)
}

# saving new data set
mod_data <- mod(data)
head(mod_data)
```

```
  marital_status app_mode app_order course attendance_time
prev_qualification
1 1              17       5         171    1                1

2 1              15       1         9254   1                1

3 1              1        5         9070   1                1
```

```
4 1               17       2          9773   1                1

5 2               39       1          8014   0                1

6 2               39       1          9991   0               19

  prev_qualification_grade nationality mot_qualification
fat_qualification …
1 122.0                     1           19                 12
…
2 160.0                     1           1                  3
…
3 122.0                     1           37                 37
…
4 122.0                     1           38                 37
…
5 100.0                     1           37                 38
…
6 133.1                     1           37                 37
…
  fst_sem_approved fst_sem_grade fst_sem_no_eval snd_sem_eval
snd_sem_grade
1 0                 0.00000      0               0             0.00000

2 6                14.00000      0               6            13.66667

3 0                 0.00000      0               0             0.00000

4 6                13.42857      0               10           12.40000

5 5                12.33333      0               6            13.00000

6 5                11.85714      0               17           11.50000

  snd_sem_no_eval unemployment_rate inflation_rate GDP   target
1 0               10.8               1.4            1.74 Dropout
2 0               13.9              -0.3            0.79 Graduate
3 0               10.8               1.4            1.74 Dropout
4 0                9.4              -0.8           -3.12 Graduate
5 0               13.9              -0.3            0.79 Graduate
6 5               16.2               0.3           -0.92 Graduate
```

Next, I am adding a unique ID column made up of the row numbers just so the students will be more identifiable.

```r
# adding a unique ID column
mod_data <- mod_data %>%
mutate(row_num = row_number()) %>%
relocate(row_num)
```

```
head(mod_data)
```

```
  row_num marital_status app_mode app_order course attendance_time
1 1       1              17       5         171    1
2 2       1              15       1         9254   1
3 3       1              1        5         9070   1
4 4       1              17       2         9773   1
5 5       2              39       1         8014   0
6 6       2              39       1         9991   0
  prev_qualification prev_qualification_grade nationality
mot_qualification …
1 1                  122.0                    1           19
…
2 1                  160.0                    1           1
…
3 1                  122.0                    1           37
…
4 1                  122.0                    1           38
…
5 1                  100.0                    1           37
…
6 19                 133.1                    1           37
…
  fst_sem_approved fst_sem_grade fst_sem_no_eval snd_sem_eval
snd_sem_grade
1 0                 0.00000      0               0             0.00000

2 6                14.00000      0               6            13.66667

3 0                 0.00000      0               0             0.00000

4 6                13.42857      0               10           12.40000

5 5                12.33333      0               6            13.00000

6 5                11.85714      0               17           11.50000

  snd_sem_no_eval unemployment_rate inflation_rate GDP   target
1 0               10.8                1.4            1.74 Dropout
2 0               13.9               -0.3            0.79 Graduate
3 0               10.8                1.4            1.74 Dropout
4 0                9.4               -0.8           -3.12 Graduate
5 0               13.9               -0.3            0.79 Graduate
6 5               16.2                0.3           -0.92 Graduate
```

I will move onto the next part of the section.

2.2. Numerical and Visual Summary: In this section, I will split up the data into the train and test sets. **I am splitting the training data by those who have either graduated or dropped out (82% of the original data) and the testing data by those who are still enrolled (18% of the original**

data). For the testing data, I will drop the outcome variable (target). And then, using the training set, I will summarize the data both numerically and visually.

```
# splitting data up into training and testing

# making this reproducible
set.seed(756)

# training data will have all of the students who have either
graduated or dropped out
training <- mod_data %>%
filter(target %in% c("Graduate", "Dropout"))

# testing data will have all the students who are currently enrolled
testing <- mod_data %>%
filter(target == 'Enrolled') %>%
select(-target)

head(training)
head(testing)
```

```
  row_num marital_status app_mode app_order course attendance_time
1 1       1              17       5         171    1
2 2       1              15       1         9254   1
3 3       1              1        5         9070   1
4 4       1              17       2         9773   1
5 5       2              39       1         8014   0
6 6       2              39       1         9991   0
  prev_qualification prev_qualification_grade nationality
mot_qualification …
1 1                  122.0                    1           19
…
2 1                  160.0                    1           1
…
3 1                  122.0                    1           37
…
4 1                  122.0                    1           38
…
5 1                  100.0                    1           37
…
6 19                 133.1                    1           37
…
  fst_sem_approved fst_sem_grade fst_sem_no_eval snd_sem_eval
snd_sem_grade
1 0                 0.00000      0               0             0.00000

2 6                14.00000      0               6            13.66667

3 0                 0.00000      0               0             0.00000
```

```
4 6                13.42857      0               10           12.40000

5 5                12.33333      0               6            13.00000

6 5                11.85714      0               17           11.50000

  snd_sem_no_eval unemployment_rate inflation_rate GDP   target
1 0               10.8               1.4            1.74 Dropout
2 0               13.9              -0.3            0.79 Graduate
3 0               10.8               1.4            1.74 Dropout
4 0                9.4              -0.8           -3.12 Graduate
5 0               13.9              -0.3            0.79 Graduate
6 5               16.2               0.3           -0.92 Graduate

  row_num marital_status app_mode app_order course attendance_time
1 17      1              18       1         9238   1
2 20      1              1        1         9853   1
3 22      1              18       4         9556   1
4 26      1              1        1         9238   1
5 28      1              1        1         9085   1
6 30      1              17       2         9500   1
  prev_qualification prev_qualification_grade nationality
mot_qualification …
1 1                  137                      1           19
…
2 1                  140                      1           19
…
3 1                  127                      1           1
…
4 1                  151                      1           19
…
5 1                  138                      1           19
…
6 1                  127                      1           3
…
  fst_sem_eval fst_sem_approved fst_sem_grade fst_sem_no_eval
snd_sem_eval
1 10           1                12.00000      0               14

2 7            6                11.66667      0               8

3 14           7                11.43750      0               9

4 8            5                11.60000      0               12

5 9            5                12.66667      2               7

6 9            6                12.93333      0               7

  snd_sem_grade snd_sem_no_eval unemployment_rate inflation_rate GDP
```

```
1 11.00000      0               10.8               1.4             1.74
2 13.50000      0               16.2               0.3            -0.92
3 11.42500      0               12.7               3.7            -1.70
4 11.00000      0                7.6               2.6             0.32
5 13.00000      0                9.4              -0.8            -3.12
6 13.71667      0               16.2               0.3            -0.92
```

```
# viewing training data
dim(training)
glimpse(training)
#head(training)
summary(training)
```

```
[1] 3630   34
```

```
Rows: 3,630
Columns: 34
$ row_num                   <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
12, 13, 14…
$ marital_status            <fct> 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1,
1, 1, 1, …
$ app_mode                  <fct> 17, 15, 1, 17, 39, 39, 1, 18, 1, 1,
1, 1, 1, …
$ app_order                 <fct> 5, 1, 5, 2, 1, 1, 1, 4, 3, 1, 1, 1,
2, 1, 1, …
$ course                    <fct> 171, 9254, 9070, 9773, 8014, 9991,
9500, 9254…
$ attendance_time           <fct> 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,
1, 1, 1, …
$ prev_qualification        <fct> 1, 1, 1, 1, 1, 19, 1, 1, 1, 1, 1, 1,
1, 42, 1…
$ prev_qualification_grade <dbl> 122.0, 160.0, 122.0, 122.0, 100.0,
133.1, 142…
$ nationality               <fct> 1, 1, 1, 1, 1, 1, 1, 1, 62, 1, 1, 1,
1, 1, 1,…
$ mot_qualification         <fct> 19, 1, 37, 38, 37, 37, 19, 37, 1, 1,
38, 19, …
$ fat_qualification         <fct> 12, 3, 37, 37, 38, 37, 38, 37, 1, 19,
19, 38,…
$ mot_occupation            <fct> 5, 3, 9, 5, 9, 9, 7, 9, 9, 4, 5, 9,
4, 4, 5, …
$ admission_grade           <dbl> 127.3, 142.5, 124.8, 119.6, 141.5,
114.8, 128…
$ displaced                 <fct> 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1,
1, 1, 1, …
$ special_needs             <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, …
$ debtor                    <fct> 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,
0, 0, 0, …
$ tuition_fees_up_to_date   <fct> 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1,
```

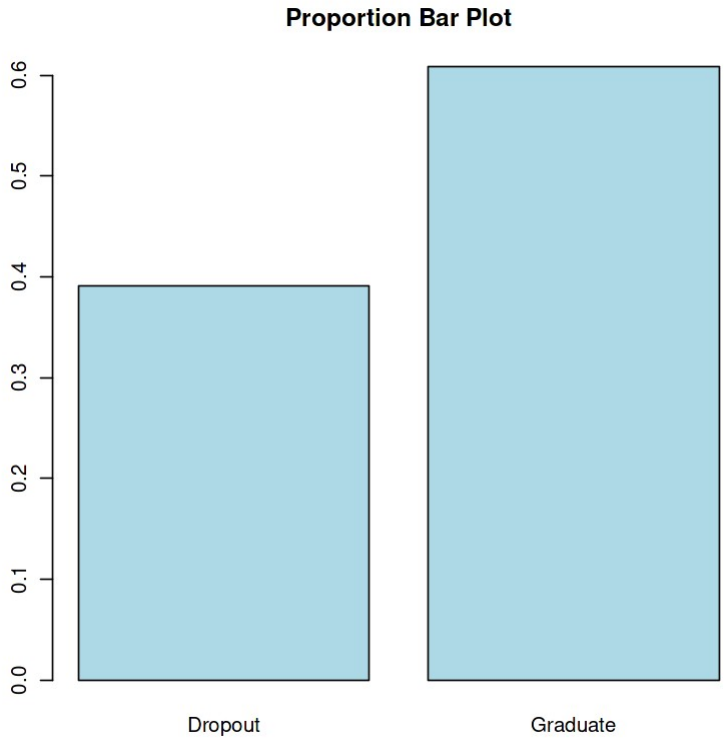
```
1, 1, 1, …
$ gender                    <fct> 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0,
0, 0, 0, …
$ scholarship_holder        <fct> 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1,
0, 1, 1, …
$ age_at_enrollment         <dbl> 20, 19, 19, 20, 45, 50, 18, 22, 21,
18, 18, 1…
$ international             <fct> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, …
$ fst_sem_credited          <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, …
$ fst_sem_enrolled          <fct> 0, 6, 6, 6, 6, 5, 7, 5, 6, 6, 6, 8,
6, 6, 5, …
$ fst_sem_eval              <fct> 0, 6, 0, 8, 9, 10, 9, 5, 8, 9, 6, 8,
6, 7, 7,…
$ fst_sem_approved          <fct> 0, 6, 0, 6, 5, 5, 7, 0, 6, 5, 6, 7,
0, 6, 4, …
$ fst_sem_grade             <dbl> 0.00000, 14.00000, 0.00000, 13.42857,
12.3333…
$ fst_sem_no_eval           <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, …
$ snd_sem_eval              <fct> 0, 6, 0, 10, 6, 17, 8, 5, 7, 14, 7,
8, 0, 8, …
$ snd_sem_grade             <dbl> 0.00000, 13.66667, 0.00000, 12.40000,
13.0000…
$ snd_sem_no_eval           <fct> 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0,
0, 0, 0, …
$ unemployment_rate         <dbl> 10.8, 13.9, 10.8, 9.4, 13.9, 16.2,
15.5, 15.5…
$ inflation_rate            <dbl> 1.4, -0.3, 1.4, -0.8, -0.3, 0.3, 2.8,
2.8, 0.…
$ GDP                       <dbl> 1.74, 0.79, 1.74, -3.12, 0.79, -0.92,
-4.06, …
$ target                    <chr> "Dropout", "Graduate", "Dropout",
"Graduate",…

    row_num     marital_status    app_mode      app_order
course
 Min.   :   1   1:3199         1      :1408   1      :2461   9500   :
666
 1st Qu.:1091   2: 327         17     : 713   2      : 451   9238   :
313
 Median :2192   3:   2         39     : 664   3      : 249   9773   :
297
 Mean   :2209   4:  75         43     : 237   4      : 218   9147   :
272
 3rd Qu.:3317   5:  22         44     : 157   5      : 129   9085   :
262
 Max.   :4424   6:   5         7      : 132   6      : 121   9670   :
220
```

```
                                (Other): 319   (Other):   1
(Other):1600
 attendance_time prev_qualification prev_qualification_grade
nationality
 0: 408          1      :3019       Min.   : 95.0
1      :3544
 1:3222          39     : 164       1st Qu.:125.0            41     :
32
                 19     : 149       Median :133.1            22     :
12
                 3      : 122       Mean   :132.9            26     :
9
                 12     :  39       3rd Qu.:140.0            6      :
8
                 40     :  34       Max.   :190.0            24     :
5
                 (Other): 103                                (Other):
20
 mot_qualification fat_qualification mot_occupation admission_grade
displaced
 1      :865       37     :1010      9      :1313   Min.   : 95.0
0:1637
 37     :840       19     : 785      4      : 670   1st Qu.:118.0
1:1993
 19     :777       1      : 732      5      : 436   Median :126.5

 38     :471       38     : 575      3      : 272   Mean   :127.3

 3      :337       3      : 220      2      : 240   3rd Qu.:135.1

 34     :127       34     : 109      7      : 224   Max.   :190.0

 (Other):213       (Other): 199      (Other): 475

 special_needs debtor   tuition_fees_up_to_date gender
scholarship_holder
 0:3590        0:3217   0: 486                  0:2381   0:2661

 1:  40        1: 413   1:3144                  1:1249   1: 969
```

```
 age_at_enrollment international fst_sem_credited fst_sem_enrolled
 Min.   :17.00     0:3544        0      :3150     6      :1594
 1st Qu.:19.00     1:  86        2      :  71     5      : 749
 Median :20.00                   1      :  67     7      : 546
 Mean   :23.46                   3      :  57     8      : 264
 3rd Qu.:25.00                   4      :  43     0      : 152
 Max.   :70.00                   6      :  43     12     :  54
                                 (Other): 199     (Other): 271
  fst_sem_eval fst_sem_approved fst_sem_grade   fst_sem_no_eval
snd_sem_eval
 8      :674   6      :1033     Min.   : 0.00   0      :3406    8
:654
 7      :621   0      : 647     1st Qu.:11.00   1      : 118    6
:556
 6      :530   5      : 530     Median :12.34   2      :  58    7
:478
 0      :321   7      : 429     Mean   :10.53   3      :  16    0
:372
 9      :296   4      : 288     3rd Qu.:13.50   4      :  14    9
:353
 10     :235   3      : 176     Max.   :18.88   6      :   5    5
:268
 (Other):953   (Other): 527                     (Other):  13
(Other):949
 snd_sem_grade   snd_sem_no_eval unemployment_rate inflation_rate
 Min.   : 0.00   0      :3416    Min.   : 7.60     Min.   :-0.800
 1st Qu.:10.52   1      : 107    1st Qu.: 9.40     1st Qu.: 0.300
 Median :12.33   2      :  35    Median :11.10     Median : 1.400
 Mean   :10.04   3      :  23    Mean   :11.63     Mean   : 1.232
 3rd Qu.:13.50   4      :  16    3rd Qu.:13.90     3rd Qu.: 2.600
 Max.   :18.57   5      :  16    Max.   :16.20     Max.   : 3.700
                 (Other):  17
      GDP               target
 Min.   :-4.060000   Length:3630
 1st Qu.:-1.700000   Class :character
 Median : 0.320000   Mode  :character
 Mean   :-0.009256
 3rd Qu.: 1.790000
 Max.   : 3.510000
```

The training set has 36 predictors and 3,630 observations. I am going to make a proportion bar plot split between 'Dropout' and 'Graduate'.

```r
# creating a proportion bar plot
prop_table <- prop.table(table(training$target))

barplot(prop_table, col = "lightblue", main = "Proportion Bar Plot")
```



It looks like about 40% of students are dropouts while the other 60% graduated.

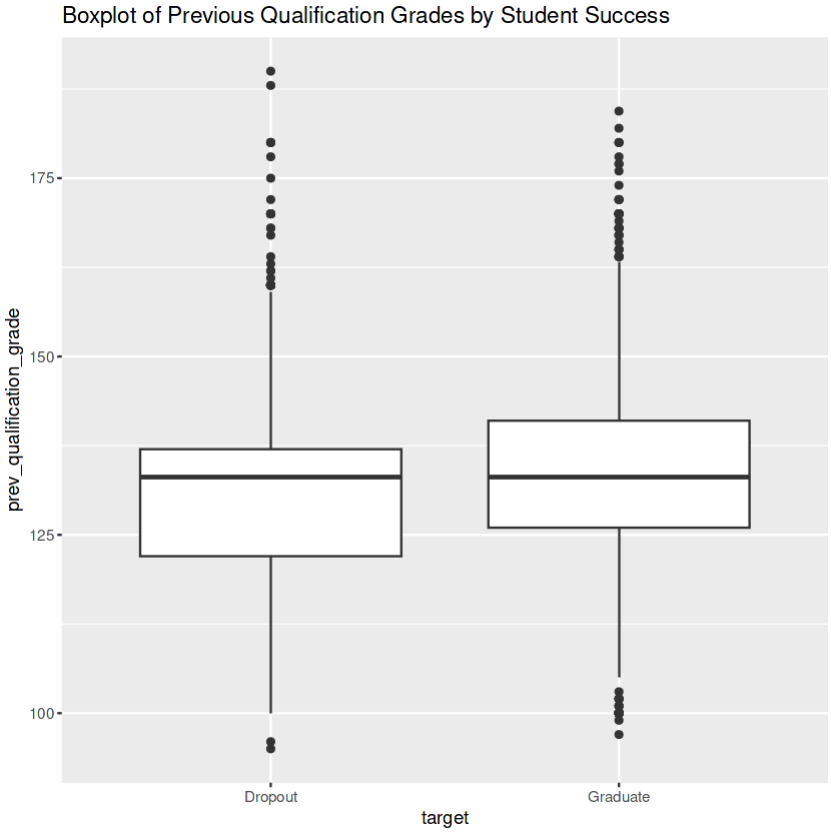
I am now making the outcome variable (target) a factor to create a couple of boxplots.

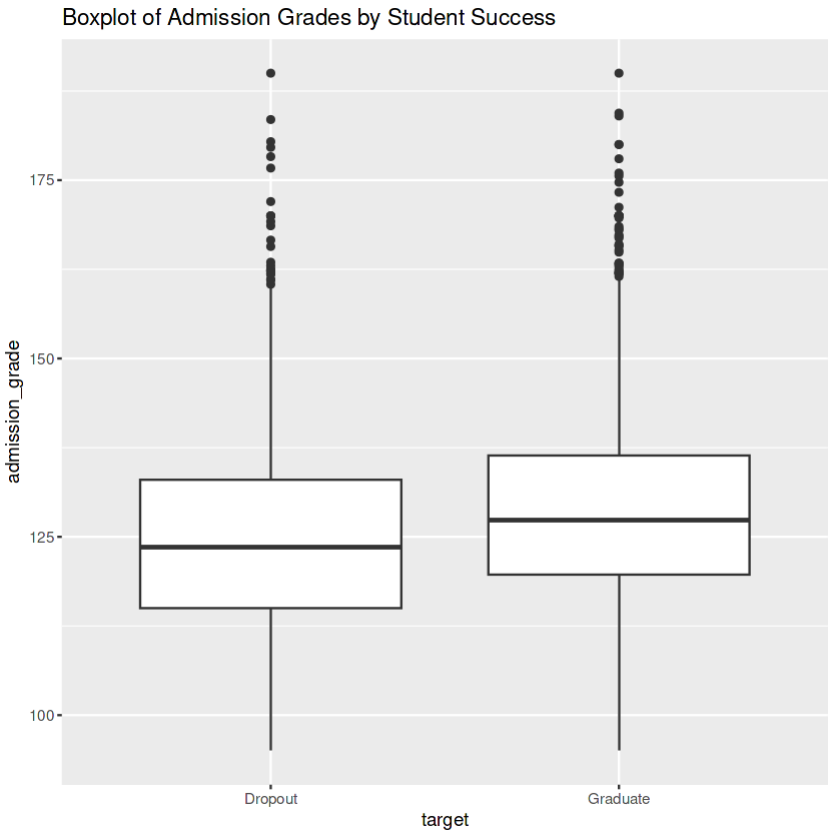
```
# making target a factor
training$target <- as.factor(training$target)
```

I am creating boxplots focusing on the outcome variable and grades (since that is most revelant to the student).

```
library(ggplot2)

# plot by previous qualification grades
ggplot(training, aes(x = target, y = prev_qualification_grade,
                     fill = prev_qualification_grade)) +
  geom_boxplot() +
  labs(title = "Boxplot of Previous Qualification Grades by Student
Success")

# plot by admission grades
ggplot(training, aes(x = target, y = admission_grade,
                     fill = admission_grade)) +
  geom_boxplot() +
  labs(title = "Boxplot of Admission Grades by Student Success")
```

```
Warning message:
“The following aesthetics were dropped during statistical
transformation: fill
ⓘ This can happen when ggplot fails to infer the correct grouping
structure in
  the data.
ⓘ Did you forget to specify a `group` aesthetic or to convert a
numerical
  variable into a factor?”
Warning message:
“The following aesthetics were dropped during statistical
transformation: fill
ⓘ This can happen when ggplot fails to infer the correct grouping
structure in
  the data.
ⓘ Did you forget to specify a `group` aesthetic or to convert a
numerical
  variable into a factor?”
```





In the first boxplot, interestingly both those who graduated and dropped out have around the same median for previous qualification grades. Additionally, dropouts have a lot of outliers with higher grades (some that are even higher than those who graduated) and also who graduated have more outliers on the bottom than those who dropped out. When it comes to the second boxplot, the median for those who graduated is slightly higher. Once again, those who dropped out have more outliers that are in a higher range than those who graduated.

Now I will be checking to see if there is any missing data that I will have to deal with.

```
# checking for missing data in training
sapply(training, function(m) sum(is.na(m)))

# checking for missing data in testing
sapply(training, function(m) sum(is.na(m)))
```

```
                  row_num           marital_status
app_mode
                        0                        0
0
               app_order                   course
attendance_time
                        0                        0
0
      prev_qualification prev_qualification_grade
nationality
```

```
                        0                         0
0
       mot_qualification        fat_qualification
mot_occupation
                        0                         0
0
         admission_grade                displaced
special_needs
                        0                         0
0
                  debtor  tuition_fees_up_to_date
gender
                        0                         0
0
      scholarship_holder        age_at_enrollment
international
                        0                         0
0
        fst_sem_credited         fst_sem_enrolled
fst_sem_eval
                        0                         0
0
        fst_sem_approved            fst_sem_grade
fst_sem_no_eval
                        0                         0
0
             snd_sem_eval            snd_sem_grade
snd_sem_no_eval
                        0                         0
0
       unemployment_rate           inflation_rate
GDP
                        0                         0
0
                   target
                        0

                  row_num           marital_status
app_mode
                        0                         0
0
                app_order                    course
attendance_time
                        0                         0
0
      prev_qualification prev_qualification_grade
nationality
                        0                         0
0
       mot_qualification        fat_qualification
```

```
mot_occupation
                        0                         0
0
         admission_grade                 displaced
special_needs
                        0                         0
0
                  debtor  tuition_fees_up_to_date
gender
                        0                         0
0
      scholarship_holder        age_at_enrollment
international
                        0                         0
0
        fst_sem_credited         fst_sem_enrolled
fst_sem_eval
                        0                         0
0
        fst_sem_approved            fst_sem_grade
fst_sem_no_eval
                        0                         0
0
            snd_sem_eval            snd_sem_grade
snd_sem_no_eval
                        0                         0
0
       unemployment_rate           inflation_rate
GDP
                        0                         0
0
                  target
                        0
```

Thankfully, **there is no missing data.** I will move onto the next section now.

3) Evaluation Metric

For my project, I will use accuracy since I only have two classes. It provides an easy interpretation for classification as it just measures the overall correctness of the predictions. Additionally, I am familiar with this evaluation metric, so it will be easier to work with.

4) Fit models

The fit models I will be utilizing for my project are kNN, random forests, and support vector machine.

4.1) Data preprocessing

I have no missing data, so I do not need to do anything for missing values. However, I will have to scale for SVM, which I will do later on.

4.2) Choose hyperparameters; fit and test models

Before I fit and test my models, I will choose my hyperparameters:

- kNN: I will find the best k
- random forest: I will find the best mtrys and trees
- svm: I will find the best sigma and C

Reminder: I am creating subsets from the training data set because my testing set does not have an outcome variable (target) to compare the predictions!

```
# subsetting
num <- length(training$row_num)

# getting splitting sample for training
train_sam <- sample(num, num / 2)
test_sam <- c(1:num)[! c(1:num) %in% train_sam]

# setting seed again
set.seed(756)

library(caret)
library(mda)
library(MLmetrics)
library(nnet)
library(readr)

# kNN
knnM <- train(target ~ marital_status + app_mode + app_order + course
+ attendance_time +
prev_qualification_grade + nationality + mot_qualification +
fat_qualification +
mot_occupation + admission_grade + displaced + special_needs + debtor
+ tuition_fees_up_to_date +
gender + scholarship_holder + age_at_enrollment + international +
fst_sem_credited +
fst_sem_enrolled + fst_sem_eval + fst_sem_approved + fst_sem_grade +
fst_sem_no_eval +
snd_sem_eval + snd_sem_grade + snd_sem_no_eval + unemployment_rate +
inflation_rate + GDP,
              data = training, subset = train_sam, method = "knn",
              trControl = trainControl(method = "repeatedcv", number =
10, repeats = 3),
              tuneGrid = data.frame(k = c(10:25)))

knnM

# saving the best model
bestK <- knnM$bestTune

# creating confusion matrix
```

```r
knn_pred_train <- predict(knnM, newdata = training, subset =
train_sam)
knnTrain <- confusionMatrix(knn_pred_train, training$target)

# extracting accuracy
accuracy <- knnTrain$overall["Accuracy"]
cat("Accuracy: ", accuracy, "\n") # making it neat

# checking F1 score
f1Score <- F1_Score(knn_pred_train, training$target)
cat("F1 Score: ", f1Score, "\n") # making it neat

# do the same thing for test set that I created
# creating confusion matrix
knn_pred_test <- predict(knnM, newdata = training, subset = test_sam)
knnTest <- confusionMatrix(knn_pred_test, training$target)

# extracting accuracy
accuracy <- knnTest$overall["Accuracy"]
cat("Accuracy: ", accuracy, "\n") # making it neat

# checking F1 score
f1Score <- F1_Score(knn_pred_test, training$target)
cat("F1 Score: ", f1Score, "\n") # making it neat
```

```
Loading required package: lattice


Attaching package: ‘caret’


The following object is masked from ‘package:purrr’:

    lift


The following object is masked from ‘package:httr’:

    progress


Loading required package: class

Loaded mda 0.5-4



Attaching package: ‘MLmetrics’


The following objects are masked from ‘package:caret’:
```

```
    MAE, RMSE


The following object is masked from ‘package:base’:

    Recall



k-Nearest Neighbors

3630 samples
  31 predictor
   2 classes: 'Dropout', 'Graduate'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 1634, 1633, 1633, 1633, 1634, 1634, ...
Resampling results across tuning parameters:

  k   Accuracy   Kappa
  10  0.7847409  0.5189332
  11  0.7858428  0.5200227
  12  0.7851163  0.5192496
  13  0.7865855  0.5216914
  14  0.7823599  0.5121042
  15  0.7840052  0.5150742
  16  0.7829144  0.5122458
  17  0.7827292  0.5120209
  18  0.7827404  0.5113859
  19  0.7786908  0.5013016
  20  0.7761237  0.4951289
  21  0.7779663  0.4988478
  22  0.7772317  0.4971207
  23  0.7763068  0.4947102
  24  0.7783286  0.4989660
  25  0.7805345  0.5028939

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 13.

Accuracy:  0.800551
F1 Score:  0.6926995
Accuracy:  0.800551
F1 Score:  0.6926995
```

My best hyperparameter for k is 13! The accuracy of the kNN model was about 80%.

Now I will move onto completing the random forest classification.

```r
# setting seed again
set.seed(756)

library(randomForest)
library(ggplot2)
library(ranger)
library(readr)
library(yardstick)
library(workflows)
library(rsample)
library(parsnip)
library(dials)
library(tidymodels)

# random forest
rf <- rand_forest() %>%
set_engine("ranger") %>%
set_mode("classification")

# create metrics
train_metrics <- metric_set(yardstick::accuracy, yardstick::f_meas)

# tuning
tune_tm <- rand_forest(trees = tune(), mtry = tune()) %>%
set_engine("ranger") %>%
set_mode("classification")

# creating the grid
rf_grid <- grid_regular(
trees(range = c(400, 1000)),
mtry(range = c(2, 10)),
levels = 5)

# creating folds for cross validation
cv_folds <- vfold_cv(training, v = 3)

# creating the workflow
rf_wf <- workflow() %>%
add_model(tune_tm) %>%
add_formula(target ~ marital_status + app_mode + app_order + course +
attendance_time +
prev_qualification_grade + nationality + mot_qualification +
fat_qualification +
mot_occupation + admission_grade + displaced + special_needs + debtor
+ tuition_fees_up_to_date +
gender + scholarship_holder + age_at_enrollment + international +
fst_sem_credited +
fst_sem_enrolled + fst_sem_eval + fst_sem_approved + fst_sem_grade +
fst_sem_no_eval +
snd_sem_eval + snd_sem_grade + snd_sem_no_eval + unemployment_rate +
```

```
inflation_rate + GDP)

# results
results <- rf_wf %>%
tune_grid(resamples = cv_folds,
          grid = rf_grid,
          metrics = train_metrics)

# visualize the results
results %>%
collect_metrics() %>%
filter(.metric == "f_meas") %>%
select(mtry, trees, mean) %>%
ggplot(aes(mtry, trees, fill = mean)) +
geom_tile()

# pull out best parameters
best_params <- results %>%
select_best("f_meas")

best_params
```

```
randomForest 4.6-14

Type rfNews() to see new features/changes/bug fixes.


Attaching package: ‘randomForest’


The following object is masked from ‘package:dplyr’:

    combine


The following object is masked from ‘package:ggplot2’:

    margin


Attaching package: ‘ranger’


The following object is masked from ‘package:randomForest’:

    importance


Attaching package: ‘yardstick’
```

```
The following objects are masked from ‘package:caret’:

    precision, recall, sensitivity, specificity

The following object is masked from ‘package:readr’:

    spec


Attaching package: ‘parsnip’

The following object is masked from ‘package:mda’:

    mars

Loading required package: scales

Attaching package: ‘scales’

The following object is masked from ‘package:purrr’:

    discard

The following object is masked from ‘package:readr’:

    col_factor


── Attaching packages ──────────────────────────────────────
tidymodels 1.1.1 ──

✔ broom        1.0.5     ✔ recipes      1.0.8
✔ infer        1.0.5     ✔ tune         1.1.2
✔ modeldata    1.2.0     ✔ workflowsets 1.0.1

── Conflicts ─────────────────────────────────────────
tidymodels_conflicts() ──
✖ randomForest::combine()  masks dplyr::combine()
✖ scales::discard()        masks purrr::discard()
✖ dplyr::filter()          masks stats::filter()
✖ recipes::fixed()         masks stringr::fixed()
✖ dplyr::lag()             masks stats::lag()
✖ caret::lift()            masks purrr::lift()
✖ randomForest::margin()   masks ggplot2::margin()
```

```
✖ parsnip::mars()           masks mda::mars()
✖ yardstick::precision()    masks caret::precision()
✖ yardstick::recall()       masks caret::recall()
✖ yardstick::sensitivity()  masks caret::sensitivity()
✖ yardstick::spec()         masks readr::spec()
✖ yardstick::specificity()  masks caret::specificity()
✖ recipes::step()           masks stats::step()
• Dig deeper into tidy modeling with R at https://www.tmwr.org

→ A | error:   User interrupt or internal error.
```

The best hyperparameters are: mtry with 4 and trees with 700! Since the mean is not on the edge of the mtry or the number of trees, I will not continue to search for the best hyperparameters because it seems that I found them already.

```
# setting seed again
set.seed(756)

# subsetting
num <- length(training$row_num)

# getting splitting sample for training
train_sam <- sample(num, num / 2)
test_sam <- c(1:num)[! c(1:num) %in% train_sam]

# getting the subsets
rf_sub_train <- training[subset = train_sam]
rf_sub_test <- training[subset = test_sam]

# specifying
rf <- rand_forest() %>%
set_engine("ranger") %>%
set_mode("classification")

# fitting training
rf_fit <- rf %>%
  fit(target ~ marital_status + app_mode + app_order + course +
attendance_time +
prev_qualification_grade + nationality + mot_qualification +
fat_qualification +
mot_occupation + admission_grade + displaced + special_needs + debtor
+ tuition_fees_up_to_date +
gender + scholarship_holder + age_at_enrollment + international +
fst_sem_credited +
fst_sem_enrolled + fst_sem_eval + fst_sem_approved + fst_sem_grade +
fst_sem_no_eval +
snd_sem_eval + snd_sem_grade + snd_sem_no_eval + unemployment_rate +
inflation_rate + GDP,
      data = rf_sub_train)
```

```r
# predict using regular training set
train_pred <- rf_fit %>%
  predict(rf_sub_train) %>%
  bind_cols(rf_sub_train) %>%
  select(target, .pred_class)

# now I will predict on the test set
test_pred <- rf_fit %>%
  predict(rf_sub_test) %>%
  bind_cols(rf_sub_test) %>%
  select(target, .pred_class)

# create metrics
train_metrics <- metric_set(yardstick::accuracy, yardstick::f_meas)

# seeing how accurate train and test are
train_pred %>%
    train_metrics(truth = target, estimate = .pred_class)

test_pred %>%
    train_metrics(truth = target, estimate = .pred_class)
```

The accurary of the random forest model is about 97%, which is pretty good!

I will now move onto SVM.

```r
# saving the outcome to add back later
train_target <- training$target

# removing target outcome from dataset
X_train <- training %>%
select(-target)

# making all of the variables numeric
X_train <- as.data.frame(lapply(X_train, as.numeric))
X_test <- as.data.frame(lapply(testing, as.numeric))

# viewing data
#head(X_train)
#head(X_test)

# scaling

# saving row_num for later
test_ID <- testing[["row_num"]]

# training
X_train_scaled <- scale(X_train[2:33])
```

```r
# testing
X_test_scaled <- scale(X_test[2:33],
                       center = attributes(X_train_scaled)
$`scaled:center`,
                       scale = attributes(X_train_scaled)
$`scaled:scale`)

#head(X_train_scaled)
#head(X_test_scaled)

# making sure X_train_scale is a data frame
X_train_scaled <- as.data.frame(X_train_scaled)

# adding target back to data
X_train_scaled <- X_train_scaled %>%
  mutate(target = as.numeric(train_target))

head(X_train_scaled)

# setting seed again
set.seed(756)

# svm, following what Joanne in Assignment 5
# specify SVM model
library(caret)
library(e1071)

# making target a factor again because it needs to be
X_train_scaled$target <- as.factor(X_train_scaled$target)

svm <- svm(target ~ marital_status + app_mode + app_order + course +
attendance_time +
prev_qualification_grade + nationality + mot_qualification +
fat_qualification +
mot_occupation + admission_grade + displaced + special_needs + debtor
+ tuition_fees_up_to_date +
gender + scholarship_holder + age_at_enrollment + international +
fst_sem_credited +
fst_sem_enrolled + fst_sem_eval + fst_sem_approved + fst_sem_grade +
fst_sem_no_eval +
snd_sem_eval + snd_sem_grade + snd_sem_no_eval + unemployment_rate +
inflation_rate + GDP,
           data = X_train_scaled)

# cross validation
ctrl <- trainControl(method = "cv", number = 5)

# creating grid
param_grid <- expand.grid(C = c(0.1, 1, 10, 100, 1000, 10000),
                          sigma = c(0.0001, 0.001, 0.01, 0.5, 1, 2, 3,
```

```
4))

tune_out <- train(target ~ marital_status + app_mode + app_order +
course + attendance_time +
prev_qualification_grade + nationality + mot_qualification +
fat_qualification +
mot_occupation + admission_grade + displaced + special_needs + debtor
+ tuition_fees_up_to_date +
gender + scholarship_holder + age_at_enrollment + international +
fst_sem_credited +
fst_sem_enrolled + fst_sem_eval + fst_sem_approved + fst_sem_grade +
fst_sem_no_eval +
snd_sem_eval + snd_sem_grade + snd_sem_no_eval + unemployment_rate +
inflation_rate + GDP,
                  data = X_train_scaled,
                  method = "svmRadial",
                  trControl = ctrl,
                  tuneGrid = param_grid,
                  metric = "F1")

tune_out
```

My hyperparameters for sigma and C are 0.001 and 1000! Now to view the accuracy.

```
# setting seed again
set.seed(756)

# subsetting
num <- length(X_train_scaled$target)

# getting splitting sample for training
train_sam <- sample(num, num / 2)
test_sam <- c(1:num)[!c(1:num) %in% train_sam]

# getting the subsets
svm_sub_train <- X_train_scaled[train_sam, ]
svm_sub_test <- X_train_scaled[test_sam, ]

library(parsnip)
library(yardstick)

# specifying
svm_mod <- svm_rbf() %>%
  set_engine("kernlab") %>%
  set_mode("classification") %>%
  set_args(cost = 100, gamma = 0.5)

# outcomes need to be factors
svm_sub_train$target <- as.factor(svm_sub_train$target)
svm_sub_test$target <- as.factor(svm_sub_test$target)
```

```r
# fit model to training data
svm_fit <- svm_mod %>%
  fit(target ~ marital_status + app_mode + app_order + course +
attendance_time +
prev_qualification_grade + nationality + mot_qualification +
fat_qualification +
mot_occupation + admission_grade + displaced + special_needs + debtor
+ tuition_fees_up_to_date +
gender + scholarship_holder + age_at_enrollment + international +
fst_sem_credited +
fst_sem_enrolled + fst_sem_eval + fst_sem_approved + fst_sem_grade +
fst_sem_no_eval +
snd_sem_eval + snd_sem_grade + snd_sem_no_eval + unemployment_rate +
inflation_rate + GDP,
      data = svm_sub_train)

# predict using regular training set
train_pred <- svm_fit %>%
  predict(svm_sub_train) %>%
  bind_cols(svm_sub_train) %>%
  select(target, .pred_class)

# predict using test data
test_pred <- svm_fit %>%
  predict(svm_sub_test) %>%
  bind_cols(svm_sub_test) %>%
  select(target, .pred_class)

# setting metrics
my_metrics <- metric_set(yardstick::accuracy, yardstick::f_meas)

# viewing training prediction
train_pred %>%
    my_metrics(truth = target, estimate = .pred_class)

# viewing test prediction
test_pred %>%
    my_metrics(truth = target, estimate = .pred_class)
```

The accuracy of the svm model is about 0.84.

5) Compare Models

- overfitting vs underfitting:
  - kNN: Both training and testing models have about the same accuracy level (80%) and Marco F1 score (0.69). It is neither overfitting or underfitting.
  - random forest: Like knn, the random forest classifier was neither overfitting or underfitting. The training and testing models had about the same accuracy level (97%) and Marco F1 score (0.96).

    - svm: SVM unfortunately was overfitting as its training data set had the accuracy of 99% and a Macro F1 score of 0.99 while the testing set had an accuracy of 82% and a Macro F1 score of 0.78.
- bias vs variance tradeoff:
    - kNN: Since this model did not experience overfitting or underfitting, it seemingly had low bias and low variance!
    - random forest: The random forest classifier also did not have overfitting or underfitting; therefore, it had low bias and low variance as well.
    - svm: The overfitting in the svm model indicates that there was low bias and high variance.
- flexibility vs interpretability:
    - Based on the book used in class, kNN, random forest, and svm are all quite flexible; however, are rather difficult to interpret. Random Forest does have provide some interpretation more so than the others, however. It allows users to plot the variance importance if they wanted to.

Random Forest was the most accurate model, so I will be using that on the testing data.

```
# following what I did before by specifying
rf <- rand_forest() %>%
  set_engine("ranger") %>%
  set_mode("classification")

# fitting on the original training data
rf_fit <- rf %>%
  fit(target ~ marital_status + app_mode + app_order + course +
attendance_time +
prev_qualification_grade + nationality + mot_qualification +
fat_qualification +
mot_occupation + admission_grade + displaced + special_needs + debtor
+ tuition_fees_up_to_date +
gender + scholarship_holder + age_at_enrollment + international +
fst_sem_credited +
fst_sem_enrolled + fst_sem_eval + fst_sem_approved + fst_sem_grade +
fst_sem_no_eval +
snd_sem_eval + snd_sem_grade + snd_sem_no_eval + unemployment_rate +
inflation_rate + GDP,
      data = training)

# predicting with test data
test_pred <- rf_fit %>%
  predict(testing) %>%
  bind_cols(testing) %>%
  select(.pred_class)

# creating a new dataframe
df <- data.frame(row_num = testing$row_num, target = test_pred)

# renaming
```

```r
df <- df %>% rename(target = colnames(df)[2])

# viewing data
df <- select(df, row_num, target)
dim(df)
head(df, 10)
```

Now, I can see which students are likely to graduate or drop out!

6) Ethical implications

My one ethical concern if this system was deployed is professors or the institution having an unconscious bias against students who are predicted to drop out. I fear they may act on this bias and not provide the proper support for their student and give up early on this student. I believe one way to combat this is by institutions providing proper training on unconscious bias and reiterating that this prediction model is to help those who may dropout succeed.