Data Mining Regressions

Michael Rose

4/24/2019

Abstract

Data

Overview of Data

The idea behind this dataset is to predict admissions into a Masters degree program. It was sampled from Engineering students at an Indian university. The parameters are the following:

parameter	range	description
GRE Score	0-340	Score on GRE exam
TOEFL Score	0 - 120	Score on TOEFL exam
University Ranking	0 / 5	Indian University Ranking
Statement of Purpose	0 / 5	Self assessed SOP score
Letter of Reccommendation	0 / 5	Self assessed LOR score
Undergraduate GPA	0 / 10	Cumulative undergraduate GPA
Research Experience	0 or 1	1 if Student engaged in research, 0 otherwise
Chance of Admit	$x \in [0, 1]$	Likelihood of admission

The source of this data is the following:

`A Comparison of Regression Models for Prediction of Graduate Admissions`

`Mohan S Acharya, Asfia Armaan, Aneeta S Antony`

`IEEE International Conference on Computational Intelligence in Data Science 2019`

Load Data

```
# read data
admissions <- read_csv(data_location,
                       # coerce data types
                       col_types = list(col_integer(), col_integer(), col_integer(), col_integer(), col

# rename features
admissions %>%
  rename("Student" = "Serial No.",
         "GRE" = "GRE Score",
```

```
         "TOEFL" = "TOEFL Score",
         "Rating" = "University Rating",
         "GPA" = "CGPA",
         "Chance" = "Chance of Admit") -> admissions
```

Visualizations

Individual Features

```
admissions
```

```
## # A tibble: 400 x 9
##    Student   GRE TOEFL Rating   SOP   LOR   GPA Research Chance
##      <int> <int> <int>  <int> <dbl> <dbl> <dbl> <fct>     <dbl>
##  1       1   337   118      4   4.5   4.5  9.65 1          0.92
##  2       2   324   107      4   4     4.5  8.87 1          0.76
##  3       3   316   104      3   3     3.5  8    1          0.72
##  4       4   322   110      3   3.5   2.5  8.67 1          0.8
##  5       5   314   103      2   2     3    8.21 0          0.65
##  6       6   330   115      5   4.5   3    9.34 1          0.9
##  7       7   321   109      3   3     4    8.2  1          0.75
##  8       8   308   101      2   3     4    7.9  0          0.68
##  9       9   302   102      1   2     1.5  8    0          0.5
## 10      10   323   108      3   3.5   3    8.6  0          0.45
## # ... with 390 more rows
```

```
# grab colnames
admissions %>% select(-c("Student", "Research")) %>% colnames() -> adm_colnames

adm_colnames
```

```
## [1] "GRE"    "TOEFL"  "Rating" "SOP"    "LOR"    "GPA"    "Chance"
```

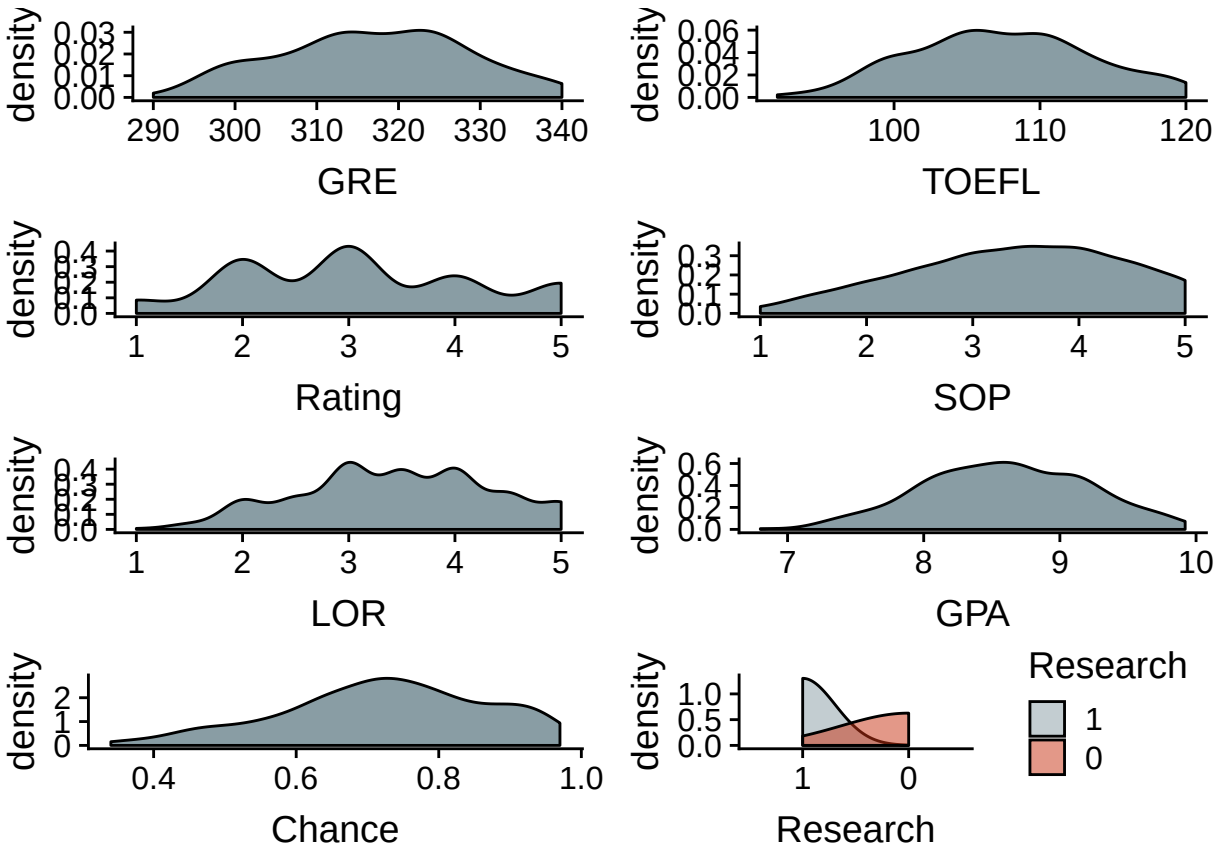
```
# make plotting function
plot_density <- function(variable){
  admissions %>%
    ggplot() +
      geom_density(aes(x = !!sym(variable)), fill = color_scheme[1])
}

# get density plots for each variable
density_plots <- future_map(adm_colnames, ~plot_density(.x))

# make a special density plot for binary Research variable
admissions %>%
  ggplot() +
  geom_density(aes(x = Research, fill = Research), alpha = 0.5) +
  scale_fill_manual(values = color_scheme) -> density_plots[[8]]

# plot
```

```
density_plots %>%
    plot_grid(plotlist = ., ncol = 2)
```

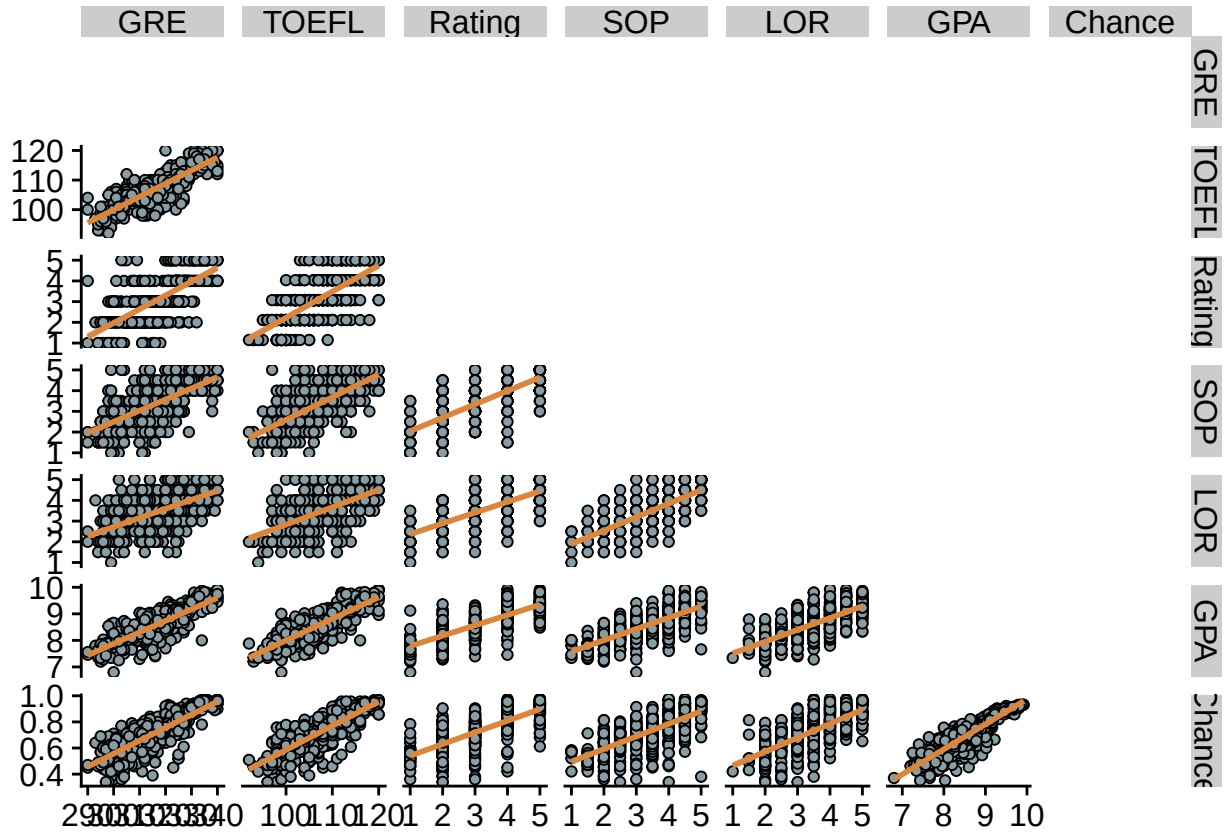


Combination of Predictors

In order to loop through each combination of predictors, I will need to nest my maps.

```
# make plotting function
plot_points <- function(data, mapping, ...){
  data %>%
    ggplot(mapping = mapping) +
    geom_point(fill = color_scheme[1], color = "black", pch = 21) +
    geom_smooth(method = "gam", color = color_scheme[4])
}

admissions %>%
  select(-c("Student", "Research")) %>%
  ggpairs(upper = "blank", diag = "blank",
          lower = list(continuous = plot_points), progress = FALSE)
```




```
admissions
```

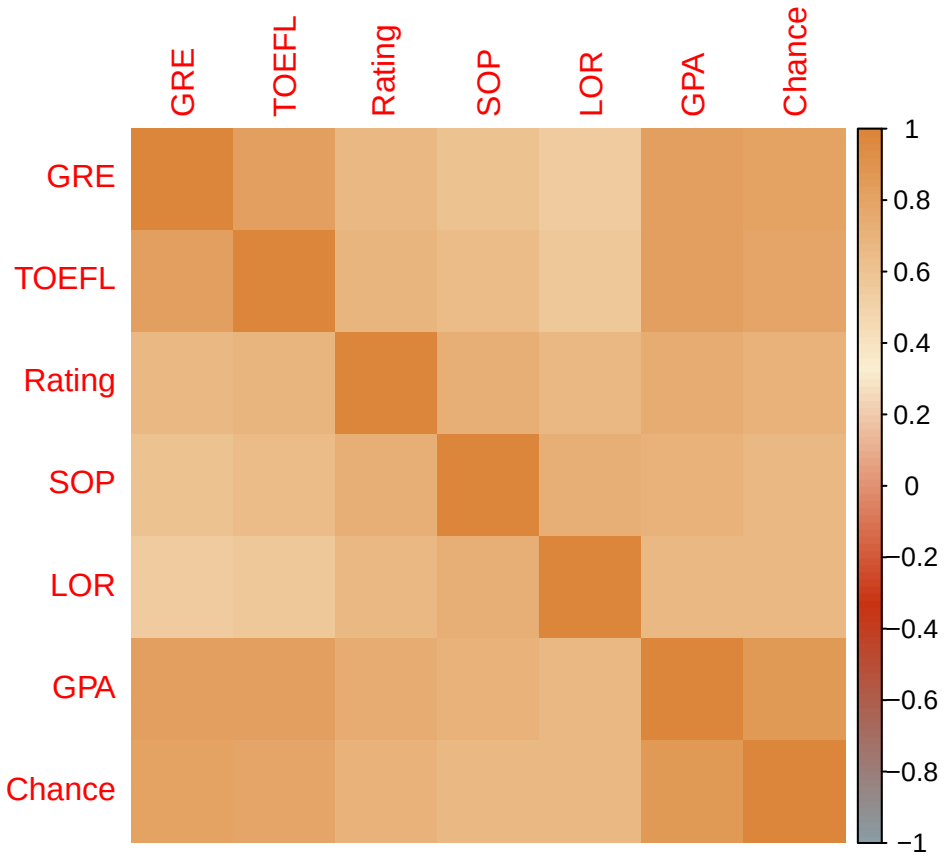
```
## # A tibble: 400 x 9
##    Student   GRE TOEFL Rating   SOP   LOR   GPA Research Chance
##      <int> <int> <int>  <int> <dbl> <dbl> <dbl> <fct>     <dbl>
##  1       1   337   118      4   4.5   4.5  9.65 1          0.92
##  2       2   324   107      4   4     4.5  8.87 1          0.76
##  3       3   316   104      3   3     3.5  8    1          0.72
##  4       4   322   110      3   3.5   2.5  8.67 1          0.8
##  5       5   314   103      2   2     3    8.21 0          0.65
##  6       6   330   115      5   4.5   3    9.34 1          0.9
##  7       7   321   109      3   3     4    8.2  1          0.75
##  8       8   308   101      2   3     4    7.9  0          0.68
##  9       9   302   102      1   2     1.5  8    0          0.5
## 10      10   323   108      3   3.5   3    8.6  0          0.45
## # ... with 390 more rows
```

Correlations

```
# create color palette for corrplot
col_ramped <- colorRampPalette(color_scheme)

# select features to plot
admissions %>%
```

```
  select(-c("Student", "Research")) %>%
  cor() %>%
  corrplot(method = "shade", col = col_ramped(100))
```



We see that most of the predictor variables have relatively high correlation.

Statistics

Algorithms

Exploration

Wrapup