An Analysis on Factors Affecting Placement Status

Ilan Lipsky, Samuel Sawyer Todd, Tran Tran

Introduction

The "Campus Recruitment" dataset is a collection of data that measures the academic performance of students from secondary to college. It also provides basic demographics such as their gender, work experience, salary offered, degree type, and placement status.

Variables:

- sl_no: Serial number
- gender: Gender of the student (Male or Female)
- ssc_p: Secondary Education percentage (10th grade)
- ssc_b: Board of Education (Central/ Others)
- hsc_p: Higher Secondary Education percentage (12th grade)
- hsc_b: Board of Education (Central/ Others)
- hsc_s: Specialization in Higher Secondary Education (Science/ Commerce/Arts)
- degree_p: Degree Percentage
- degree_t: Undergraduate Degree Type (Sci&Tech/Comm&Mgmt/Other)
- workex: Work Experience (Yes/ No)
- etest_p: Employability test percentage (conducted by the college)
- specialisation: Post Graduate Specialization (Mkt&HR/Mkt&Fin)
- mba_p: MBA percentage
- status: Placement status (Not Placed/Placed)
- salary: Salary offered by corporate to candidates

The main question we are trying to answer using this dataset is:

What important factors influenced a candidate in getting recruited?

What important factors influenced a candidate in getting recruited?

We will be using two different models and several graphs to answer this question and analyze our dataset. Namely we will be using logistic regression and random forest. First, we will start with logistic regression and use Stepwise algorithm to optimize the model. Afterward, we will start with random forest and use feature importance as well as hyper-parameters tuning to optimize the model. Both models will be using placement status as a response variable to compare and contrast which predictors are of significant for a candidate to be recruited.

1. Logistic Regression Model

1.1 Model Equation and Background

First we will build the model using all of the predictors variables except serial number and salary, because having a salary mean the candidate already recruited and including it will cause the result to be inaccurate

Model equation:

$log(\frac{p}{1-p}) = \beta_0 + \beta_1 * ssc_p + \beta_2 * ssc_b + \beta_3 * hsc_p + \beta_4 * hsc_b + \beta_5 * hsc_s + \beta_6 * degree_p + \beta_7 * degree_t + \beta_8 * workex + \beta_9 * etest_p + \beta_{10} * specialisation + \beta_{11} * mba_p + \beta_{12} * gender + \epsilon$

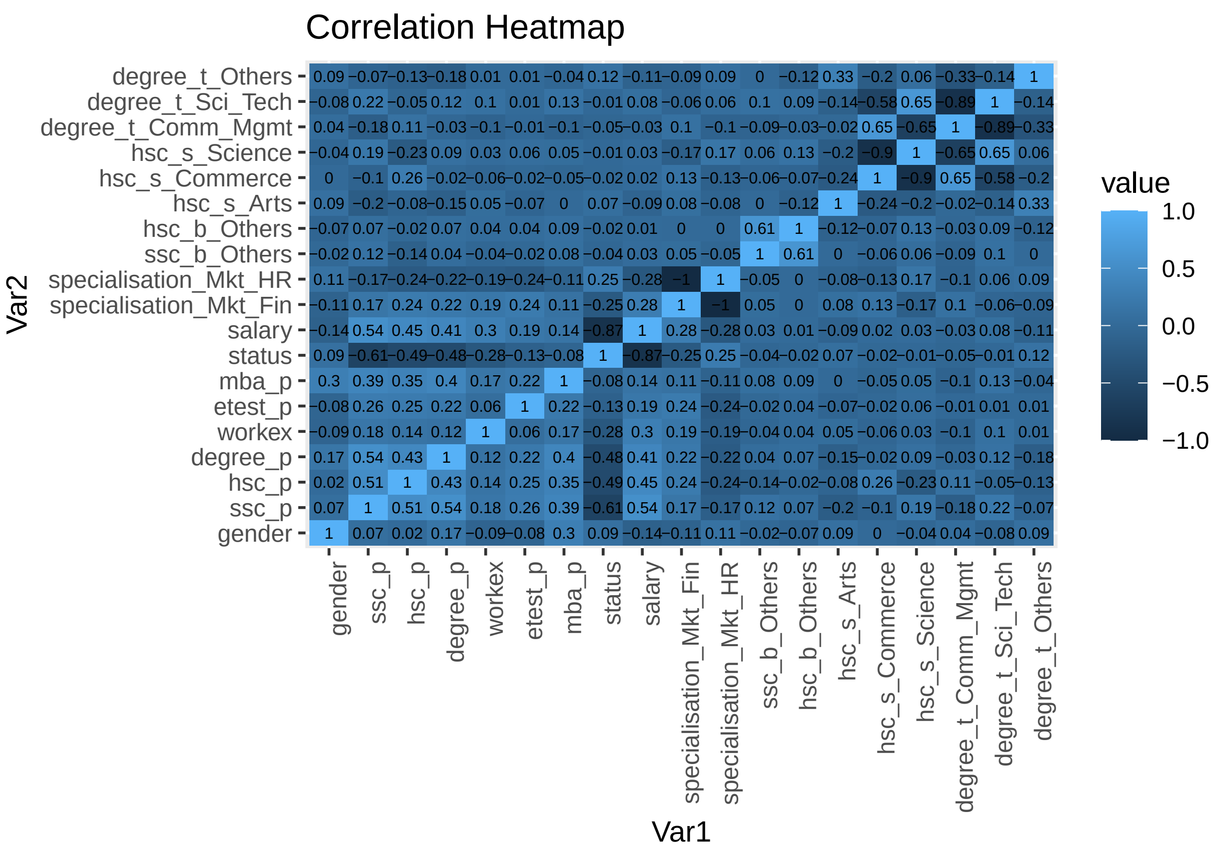
$\beta_8 = \{0 \text{ if no, } 1 \text{ if yes}\}$

$\beta_1 2 = \{0 \text{ if male, } 1 \text{ if female}\}$

Using this model equation we will look over which predictors are of significant importance using the optimization method mentioned previously. Logistic regression was used because the response variable is categorical with a binary outcome, placed or not placed. Another reason is because this model less complex but still providing a good interpretability, which align with the goal for answering the project question. Unfortunately, logistic regression is more sensitive to outliers and overfitting compare to more complex model like random forest.

Data Cleaning

Data Visualization



Count Plots

Logistic Regression

```
##
## Call:
## glm(formula = status ~ . - salary, family = "binomial", data = college_df,
##     maxit = 1000)
##
## Coefficients: (3 not defined because of singularities)
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)            19.47031    5.03022   3.871 0.000109 ***
## gender                  1.19433    0.68598   1.741 0.081673 .
## ssc_p                  -0.22891    0.04682  -4.889 1.01e-06 ***
## hsc_p                  -0.10721    0.03778  -2.838 0.004541 **
## degree_p               -0.18577    0.05558  -3.343 0.000830 ***
## workex                 -2.08385    0.70839  -2.942 0.003264 **
## etest_p                 0.01416    0.02266   0.625 0.532060
## mba_p                   0.21413    0.05852   3.659 0.000253 ***
## specialisation_Mkt_Fin -0.26381    0.55610  -0.474 0.635217
## specialisation_Mkt_HR        NA         NA      NA       NA
## ssc_b_Others           -0.22767    0.71685  -0.318 0.750787
## hsc_b_Others           -0.33074    0.73509  -0.450 0.652757
## hsc_s_Arts             -0.91121    1.45714  -0.625 0.531746
## hsc_s_Commerce          0.58666    0.78080   0.751 0.452440
## hsc_s_Science                NA         NA      NA       NA
```

```
## degree_t_Comm_Mgmt     -1.11791    1.54778  -0.722 0.470132
## degree_t_Sci_Tech       0.60785    1.67905   0.362 0.717337
## degree_t_Others              NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 266.771  on 214  degrees of freedom
## Residual deviance:  99.677  on 200  degrees of freedom
## AIC: 129.68
##
## Number of Fisher Scoring iterations: 7
```

```
##
## Call:  glm(formula = status ~ gender + ssc_p + hsc_p + degree_p + workex +
##     mba_p + degree_t_Comm_Mgmt, family = "binomial", data = college_df,
##     maxit = 1000)
##
## Coefficients:
##        (Intercept)              gender               ssc_p               hsc_p
##            19.8021              1.2342             -0.2192             -0.1019
##           degree_p              workex               mba_p  degree_t_Comm_Mgmt
##            -0.1737             -2.3669              0.1986             -1.2587
##
## Degrees of Freedom: 214 Total (i.e. Null);  207 Residual
## Null Deviance:       266.8
## Residual Deviance: 103.5     AIC: 119.5
```

```r
set.seed(41)
test_error = numeric(10)

for (i in 1:10) {
  sample_indices = sample.int(n = nrow(college_df), size = floor(0.8 * nrow(college_df)), replace = FALS
  train = college_df[sample_indices,]
  test = college_df[-sample_indices,]

  college_glm = glm(status ~ gender + ssc_p + hsc_p + degree_p + workex + mba_p + degree_t_Comm_Mgmt,
                    data = train,
                    family = "binomial")

  college_pred = predict.glm(college_glm, newdata = test, type = "response")
  yhat = ifelse(college_pred < 0.5, 'Not Placed', 'Placed')

  conf.test = table(test$status, yhat)
  test_error[i] = (conf.test[1, 2] + conf.test[2, 1]) / nrow(test)
}
mean(test_error)
```

```
## [1] 0.1139535
```