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 a Stepwise algorithm to optimize the model. Afterward, we will start with random forest and use feature importance to optimize the model. Both models will be using placement status as a response variable to compare and contrast which predictors are significant for a candidate to be recruited.

1. Logistic Regression Model (Tran Tran)

1.1 Model Equation and Background

First, we will build the model using all of the predictor variables except serial number and salary, because having a salary means the candidate has already been recruited and including it will cause the result to be interpret inaccurately.

Model equation:

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

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

$\beta_{12} = \{0 \text{ if male}, 1 \text{ if female}\}$

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 that this model is less complex but still provides good interpretability, which aligns with the goal of answering the project question. Unfortunately, logistic regression is more sensitive to outliers and overfitting compared to more complex models like random forest.

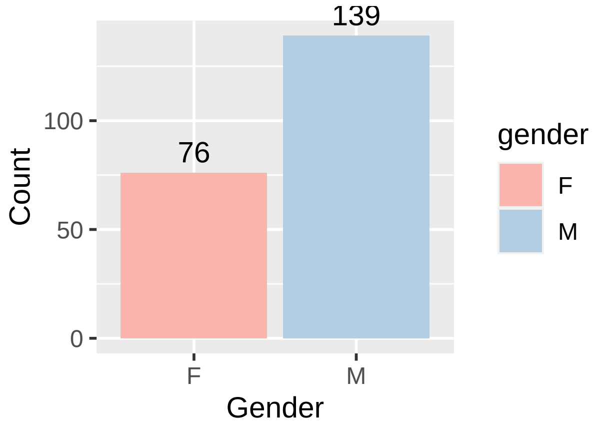
In regard to the outliers issue I have created multiple distribution plots for all of the continuous variables to see if it is a problem and luckily the outliers are very few to introduce a bias. Then for the overfitting possibility, by using the 80/20 split during training and 10 random subdivision this should not be an issue.

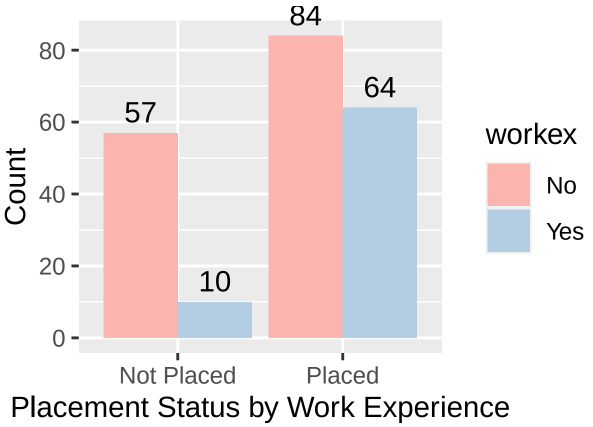
1.2 Building Logistic Regression Model

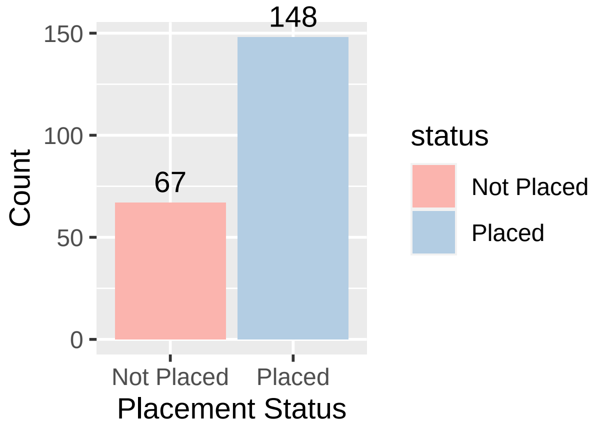
First I will run a summary report on the model with all of the predictors except salary as reasoned previously. Here I will look over which predictors are statistically significant and I can see that ssc_p, hsc_p, degree_p, workex, and mba_p P-values are less than 0.05, indicating that it is significant to our response variable, the only one need further assessing is gender because its p-value is slightly higher than the common threshold.

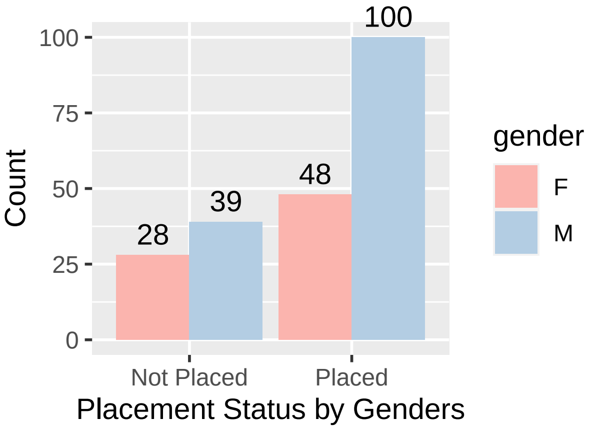
```
##
## Call:
## glm(formula = status ~ . - salary, family = "binomial", data = college_df,
##     maxit = 1000)
##
## Coefficients: (3 not defined because of singularities)
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -19.47031    5.03022  -3.871 0.000109 ***
## 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
```









Looking over the count plots above, we can see that they match some of the assumptions that were found in the summary table. Placement status by genders for example, while at first seemingly skewed toward male students being favored, we have to remember the gender population in the dataset is 76 females vs 139 males, male students are nearly double compared to female students. The placement status by gender showed that there is bias against female students in proportion to the overall gender distribution in the dataset, and while it is not a large bias, I think it is still beneficial to include it in the final building process.

Now I will run the Stepwise algorithm which will progressively try to minimize the AIC and maximize the R^2 by removing and adding predictors to the model.

```r
table = step(college_glm_1, trace = 0 )
table
```

```
##
## 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
```

The final result for the best optimization matched well with what we already saw through the summary table before running the step() function. With gender, ssc_p, hsc_p, degree_p, workex, and mba_p being the best predictors for the model, but we also see degree_t_Comm_Mgmt being the extra predictor in the result despite its high P-value. This could be because while it is not significant when combined with the initial all predictors run, it is in fact significant when we run with this selected group of predictors instead.

1.3 Logistic Regression Results

The building process is finished, so I will now perform cross validation with 80% training and 20% testing sets.

```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
```

The result is quite promising, with an average test error rate of only 11.40% for the logistic regression model. This indicates that the model performs reasonably well in predicting the outcome for placement status

```r
exams.glm2 = glm(status ~ gender + ssc_p + hsc_p + degree_p + workex + mba_p + degree_t_Comm_Mgmt,
                     data = train,
                     family = "binomial")
summary(exams.glm2)
```

```
##
## Call:
## glm(formula = status ~ gender + ssc_p + hsc_p + degree_p + workex +
##     mba_p + degree_t_Comm_Mgmt, family = "binomial", data = train)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -17.67537    4.51247  -3.917 8.97e-05 ***
## gender              -1.17240    0.62091  -1.888 0.058999 .
## ssc_p                0.21353    0.04262   5.010 5.43e-07 ***
## hsc_p                0.08233    0.03502   2.351 0.018731 *
## degree_p             0.15468    0.05309   2.914 0.003573 **
## workex               2.32904    0.72149   3.228 0.001246 **
## mba_p               -0.18880    0.05391  -3.502 0.000462 ***
## degree_t_Comm_Mgmt   1.26663    0.61693   2.053 0.040061 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 217.066  on 171  degrees of freedom
## Residual deviance:  93.491  on 164  degrees of freedom
## AIC: 109.49
##
## Number of Fisher Scoring iterations: 6
```

Looking over all of the results, we can now answer the question: "What important factors influenced a candidate in getting recruited?". From the summary table, we can see that if your gender is male, with a higher Secondary Education percentage, Higher Secondary Education percentage, Degree percentage and you have work experience as well as pursuing a degree in Communication and Management then your odds of getting recruited increase. Interestingly, having a higher MBA percentage actually decreases the log-odds of getting hired. This I surmise might be due to a diminishing return issue, because in the specialization column, there are only two specialties for students going into post-grad, which are Marketing & Finance or Marketing & HR, both of which are not exactly required going into the field, unlike more STEM-focused fields. It could also be because while having a good MBA percentage is generally beneficial, there may be a point beyond which additional percentage points do not contribute significantly to getting hired.

2. Random Forest Model (Samuel Sawyer Todd, Ilan Lipsky)

2.1 Model Equation and Background

Next we will build the Random Forest using all of the predictors variables except serial number and salary, because having a salary means the candidate is already recruited, and including it will cause the result to be an inaccurate interpretation.

Model equation:

$status \sim gender + ssc\_p + ssc\_b + hsc\_p + hsc\_b + hsc\_s + degree\_p + degree\_t + workex + etest\_p + specialisation + mba\_p$

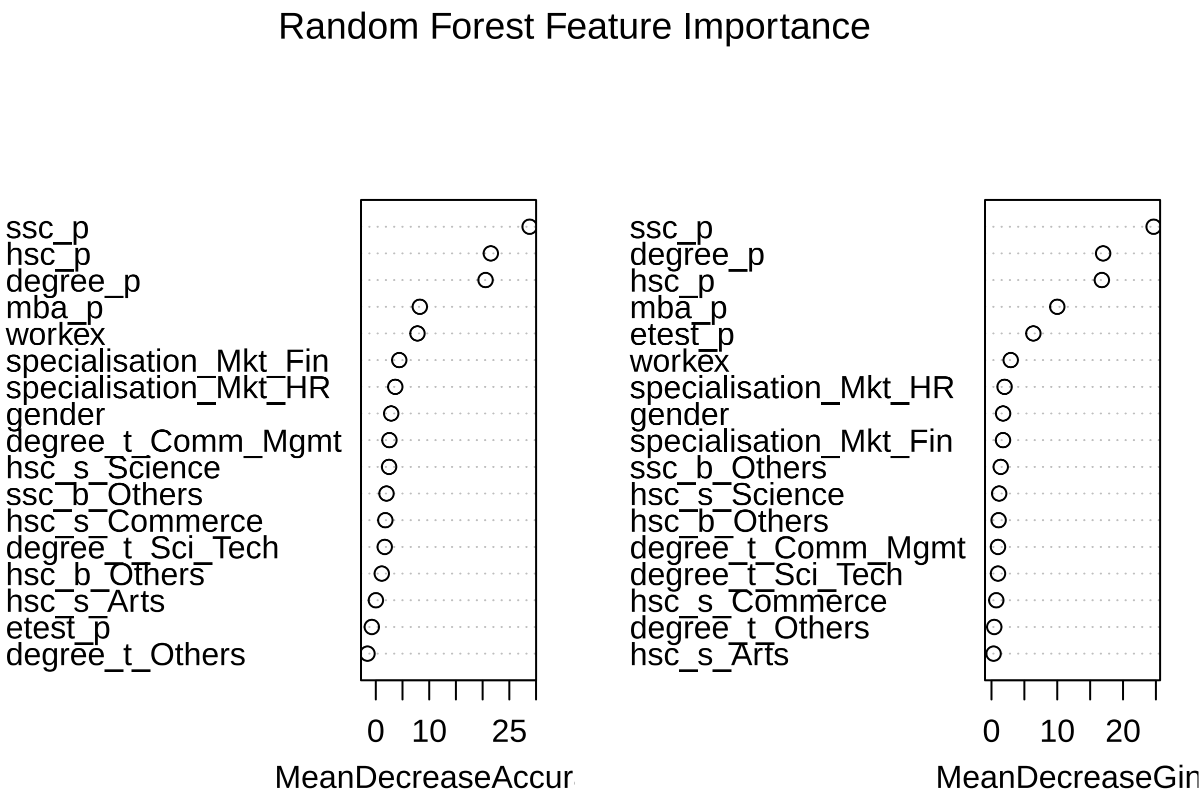
Using this model equation we will look over which predictors are of significant importance using variable importance plot. Random Forest was used because it is more robust to overfitting compared to Logistical Regression, especially when dealing with high-dimensional data. It combines predictions from multiple decision trees, providing a natural form of regularization. Additionally, it has a built-in feature importance measure, which enables the identification of key predictors in the model. However, Random Forest models are computationally intensive and may overfit to noise in training data if the number of trees is too high

After converting the binary values to factors to ensure compatibility with the random forest algorithm, we split the data set into training and testing sets using the same fixed seed that was used for our Logistical Regression.

We then utilize the randomForest function to train a random forest model in predicting admission status, excluding the "salary" variable.

```r
# Train random forest model
rf_model_importance <- randomForest(status ~ . -salary, data = college_df, importance = TRUE)

# Plot feature importance
varImpPlot(rf_model_importance, main = "Random Forest Feature Importance")
```



We can see in the MeanDecreaseGini plot graph of feature importance that "ssc_p", "hsc_p", "degree_p", "mba_p", "etest_p", and "workex" are the most significant variables in predicting college admission status as they are nodes of the steepest drop curve.

Next, we rigorously test the trained Random Forest model by applying it to the designated test set, generating predictions that serve as a critical measure of its efficacy. To ensure consistency with the original status variable, the predicted values are converted into factors, maintaining alignment with the variable's distinct levels. We then use this evaluation to compute our confusion matrix, which provides the detailed breakdown into the model's accuracy, highlighting areas of strength and areas that may require refinement.

```r
# Make predictions on the test set
rf_pred <- predict(rf_model_importance, newdata = test_data)

# Convert predicted values to factors with the same levels as the original status variable
rf_pred <- factor(rf_pred, levels = levels(test_data$status))

# Confusion matrix
conf_matrix_rf <- confusionMatrix(data = rf_pred, reference = test_data$status)
print(conf_matrix_rf)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 15  0
##          1  0 28
##
##                Accuracy : 1
##                  95% CI : (0.9178, 1)
##     No Information Rate : 0.6512
##     P-Value [Acc > NIR] : 9.742e-09
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.3488
##          Detection Rate : 0.3488
##    Detection Prevalence : 0.3488
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

2.2 Building Random Forest Model

We attempted to use hyper-parameters for Random Forest optimization, however the default parameters were chosen as they are already optimized for the model. Therefore, we decided against updating the hyper-parameters.

2.3 Random Forest Results

Lastly, the following code showcases a more robust assessment of model performance using the variables of significant importance and an iterative evaluation over 10 splits and calculations of the mean test prediction error.

```r
set.seed(41)
test_errors <- numeric(10)

for (iteration in 1:10) {
  sample_indices <- sample.int(n = nrow(college_df), size = floor(0.8 * nrow(college_df)), replace = FA
  train_data <- college_df[sample_indices, ]
  test_data <- college_df[-sample_indices, ]

  rf_model <- randomForest(status ~ ssc_p + hsc_p + degree_p + mba_p + etest_p + workex, data = train_da

  rf_pred <- predict(rf_model, newdata = test_data)

  test_error <- mean(rf_pred != test_data$status)

  test_errors[iteration] <- test_error

}

mean_test_error <- mean(test_errors)
print(mean_test_error)
```

```
## [1] 0.1744186
```

Bibliography

Roshan B. (April, 2020). Campus Recruitment. Retrieved September 30, 2023 from https://www.kaggle.com/datasets/benroshan/factors-affecting-campus-placement

Conclusion