**Padding Validation Model**

**Introduction**

In recent recruitment process, placement officer wants to pay more attention on the number of candidate they need to invite before deadline. In order to help to reach this goal, different machine learning model are built and tested. This padding validation model helps to predict the presence rate of a specific JSR based on its padding rate and other factors such as AA, Post, Sector and etc. Placement officers can come up with a potential number for inv\_at\_ddl based on their expertise and this padding validation model will calculate the padding rate and predict the final presence rate based on that padding rate. And then placement officer can decide whether they want to modify the number of inv\_at\_ddl. Historical data of the past two years (from 2016 to 2018) were used. The equation to predict the presence rate is: pres = 1.198417 – 0.286109 \* padding\_rate + AA. And based on the presence rate we can back calculate the inv\_at\_ddl by running the R code.

In the process, 2016Q3-2017Q3 (281 observations in total) are used as training set, 2017Q4-2018Q3 (249 observations in total) is used as testing set to validate the accuracy of the prediction. 95.65% of the predicted enter\_on\_duty number is within ±2 of the actual enter\_on\_duty number in the testing set and 78.26% of the predicted enter\_on\_duty number is within ±1 of the actual enter\_on\_duty number in the testing set, meaning the model is doing a good job. The details are shown as follows.

**Data Explanation**

1. Data Source

The dataset is downloaded from the DOVE, James Model 2- updated, described at a project level.

1. Variable Introduction

|  |  |
| --- | --- |
| Variable Name | Interpret |
| pres | Presence rate: calculated by enter\_on\_duty / inv\_at\_ddl |
| padding\_rate | Padding rate: calculated by inv\_at\_ddl / VR |
| fill\_rate | calculated by enter\_on\_duty / VR |
| AA | Assignment Area. Different project has different coefficient. |
| Post | Countries where the project is located. |
| VR | The number of volunteer requested by each project. |
| inv\_at\_ddl | The number of people invited at deadline. |
| enter\_on\_duty | The number of people that entered on duty at the deadline. |

**Model Construction**

Every JSR has a specific presence rate which is calculated by enter\_on\_duty / inv\_at\_ddl and it is always fall between 0 and 1. And each JSR also has a padding rate which is calculated by inv\_at\_ddl / VR and it is always greater and equal to 1. I believe there is a linear relations between the padding rate and the presence rate. To fit the equation to predict the correct presence rate, I tried to add different variables like AA, Post, and etc, and ended up keeping only the padding rate and AA which predicts the presence rate most accurately. What’s more, when looking at the summary of the model, padding rate is more than 0.9999 significant which confirm my assumption that presence rate is strongly correlated to padding rate. Some AA have significant effect, namely AA144 and AA172 are 0.9999 significant; AA145, AA175 and AA191 are 0.999 significant; AA110, AA140, AA154, and AA170 are are 0.99 significant; AA164 and AA171 are 0.95 significant.

**Validation Method**

In order to test our model, I first trained the validating model using 2016Q3-2017Q3 (281 observations in total). Then, I used 2017Q4-2018Q3 (249 observations in total) as testing set, calculated the actual padding rate, generated predicted presence rate; Using the VR divided by the predicted presence rate to get predicted inv\_at\_ddl and predicted enter\_on\_duty; Then compare the predicted enter\_on\_duty to the actual enter\_on\_duty to validate the accuracy of the prediction. The equation is pres = 1.198417 – 0.286109 \* padding\_rate + AA. 91.90% of the predicted enter\_on\_duty number is within ±2 of the actual enter\_on\_duty number in the testing set and 72.87% of the predicted enter\_on\_duty number is within ±1 of the actual enter\_on\_duty number in the testing set, meaning the validation model is doing a good job to predict the relationship between presence rate and padding rate.

Last but not least, after created the final validation model using all the data we have from 2016Q3-2018Q3 (530 observations in total). I got a small set of July 2018 departing data (23 observations) to validate the prediction. 95.65% of the predicted enter\_on\_duty number is within ±2 of the actual enter\_on\_duty number in the July 2018 dataset and 78.26% of the predicted enter\_on\_duty number is within ±1 of the actual enter\_on\_duty number in July 2018 dataset, reassuring the model’s performance of predicting presence rate based on padding rate and AA.

**Conclusion**

The final equation is using all the data we have from 2016Q3-2018Q3 (530 observations in total). The purpose of this model is to predict presence rate based on padding rate, which means it will not directly generate inv\_at\_ddl for each JSR. Placement officers need to use their area of expertise and experience to come up with inv\_at\_ddl number first and then the validation model can help predict based on this padding rate what are the presence rate and what are the predicted enter\_on\_duty; And then by add/subtract the gap between VR and predicted enter\_on\_duty to their first guess of inv\_at\_ddl, they can have a more accurate prediction of inv\_at\_ddl.

pres = 1.198417 – 0.286109 \* padding\_rate + AA

The coefficient of AA is:

|  |  |
| --- | --- |
| Assignment Area | Coefficient |
| AA103 | -0.106503 |
| AA104 | -0.016130 |
| AA110 | -0.132905 \* |
| AA114 | -0.066389 |
| AA117 | -0.055716 |
| AA124 | 0.006646 |
| AA131 | -0.142453 |
| AA134 | -0.102587 |
| AA140 | -0.125858 \* |
| AA144 | -0.238168 \*\*\* |
| AA145 | -0.162221 \*\* |
| AA154 | -0.114320 \* |
| AA155 | -0.061905 |
| AA162 | -0.054632 |
| AA164 | -0.100383 . |
| AA170 | -0.130041 \* |
| AA171 | -0.091348 . |
| AA172 | 0.240113 \*\*\* |
| AA173 | -0.087663 |
| AA175 | -0.151413 \*\* |
| AA177 | -0.030611 |
| AA191 | -0.163332 \*\* |
| AA199 | -0.112454 |

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘·’