Productivity Project

Section 1 - Introduction

Nearly every job in existence has some form of performance metric, usually centered around productivity. What productivity looks like may differ from job to job, however, they all have some way of measuring an employee's performance. If people are not meeting their expected goals for productivity, it may clue the employer into something being wrong. At worse, not meeting expectations may result in letting the employee go.

It is unsurprising, then, that companies are highly interested in and invested in the productivity of their employees. With this in mind, being able to predict if an employee is meeting their productivity goal would be incredibly useful for an employer. Being aware of potentially underperforming employees would also allow employers to intervene early to provide help to their employees and ensure they are able to meet continue meeting their goals.

The data used for this project was downloaded from the UC Irvine Machine Learning Repository and contains information on 12 teams within the garment industry. Each row provides information about the team's performance on a given day, over the course of 88 to 109 days, for a total of 1197 collective observations.

The garment industry is a highly demanding industry which has a particular focus on productivity due to its global demand (Imran et al., 2021). To meet the ever growing demand, employee productivity within this industry is particularly vital (Balla et al., 2020)

For the purposes of this analysis, I will be creating a dichotomous variable to represent whether or not a specific team met their targeted productivity. I will be attempting to predict whether or not they met their goal by looking at a number of variables included in this dataset. See below for a list of the variables included in the dataset:

- date : Date in MM-DD-YYYY
- day : Day of the Week (Monday, Tuesday, Wednesday, etc.)
- quarter : A portion of the month. A month was divided into four quarters
- department : Associated department with the instance
- team : Associated team number with the instance
- no_of_workers : Number of workers in each team

- no_of_style_change : Number of changes in the style of a particular product
- targeted_productivity : Targeted productivity set by the Authority for each team for each day.
- smv : Standard Minute Value, this is the allocated time for a task
- wip : Work in progress. Includes the number of unfinished items for products
- over_time : Represents the amount of overtime by each team in minutes
- incentive : Represents the amount of financial incentive (in BDT) that enables or motivates a particular course of action.
- idle_time : The amount of time when the production was interrupted due to several reasons
- idle_men : The number of workers who were idle due to production interruption
- actual_productivity : The actual % of productivity that was delivered by the workers. It ranges from 0-1.

Section 2 - Data

> Note about the data:
>
> Some of the variables below - specifically, the date, department, and day of the week - are non-continuous variables and will likely need to be dropped in order to perform the analyses and build the models necessary. I intend to keep the variable "quarter" as the time of month is likely to be an important factor, however, it will need to be re-coded for the purpose of this project. With dropping the previously mentioned variables, I am still over the minimum requirement of at least 10 predictors.

Table 1: Data summary

Name	employee
Number of rows	1197
Number of columns	15
Column type frequency:	
character	4
numeric	11
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
date	0	1	8	9	0	59	0
quarter	0	1	8	8	0	5	0
department	0	1	6	9	0	2	0
day	0	1	6	9	0	6	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
team	0	1.00	6.43	3.46	1.00	3.00	6.00	9.00	12.00	
targeted_productivity	0	1.00	0.73	0.10	0.07	0.70	0.75	0.80	0.80	
smv	0	1.00	15.06	10.94	2.90	3.94	15.26	24.26	54.56	
wip	506	0.58	1190.47	1837.46	7.00	774.50	1039.00	1252.50	23122.00	
over_time	0	1.00	4567.46	3348.82	0.00	1440.00	3960.00	6960.00	25920.00	
incentive	0	1.00	38.21	160.18	0.00	0.00	0.00	50.00	3600.00	
idle_time	0	1.00	0.73	12.71	0.00	0.00	0.00	0.00	300.00	
idle_men	0	1.00	0.37	3.27	0.00	0.00	0.00	0.00	45.00	
no_of_style_change	0	1.00	0.15	0.43	0.00	0.00	0.00	0.00	2.00	
no_of_workers	0	1.00	34.61	22.20	2.00	9.00	34.00	57.00	89.00	
actual_productivity	0	1.00	0.74	0.17	0.23	0.65	0.77	0.85	1.12	

Section 3 - Outcome and predictor variables

As stated previously, I am creating a dichotomous variable that indicates whether or not a particular team met their target production goals. This new variable will be called `goal_met`, with a 1 indicating the goal was met, and a 0 indicating the team fell below their targeted productivity on that day.

```
# A tibble: 2 x 2
  goal_met     n
     <dbl> <int>
1        0   322
2        1   875
```

As indicated above, teams fell below their targeted productivity 322 times. There is class imbalance in the outcome variable, which may need to be accounted for.

The majority of the variables included are likely to be important predictors for the outcome - however, there are a few that I think will be particularly important. For instance, the number of workers on a team and the incentive available to the workers for meeting goals will likely be important. The number of style changes that a particular product has is also likely to be very predictive as a higher number would indicate a more complex product and/or a constant need to adapt. More overtime work is also likely to be predictive of meeting productivity goals, as the overtime is likely to be spent ensuring that the goals are met. Idle time and idle men will also likely be very important in ensuring productivity is high.

Section 4 - Modeling plan

Specific research questions:

Does higher monetary incentives predict meeting performance goals?

- I hypothesize that incentives will be highly predictive of meeting performance goals - higher incentives for the team will increase the motivation each team feels form meeting their goals. I will examine this by looking at feature importance plots and by creating PDP plots looking at the incentive variable.

Will the number of workers on a team interact with the number of works in progress to predict performance goals being met?

- Having a high number of workers alone is likely to be good for meeting performance goals, however, looking at the interaction between the number of workers as well as the number of in progress projects will likely more accurately predict the performance of that team.

Lastly, will the number of style changes be an important predictor for meeting performance goals?

- A higher number of changes likely indicates more complex work that a particular team is doing, and it suggests a greater need for constant adaptation from that team. This constant need for adaptation and constant interruption of their current flow will likely be an important predictor for performance goals being met. I will examine this by looking at feature importance plots and by creating PDP plots looking at the number of style changes variable.

I will be examining several different models to see which models perform the best for this kind of data. For these models, I will look at two tree-based methods - random forest and xgboost - as well as a naive bayes model and a kNN model. I will evaluate the balanced accuracy and confusion matrices for all four models to compare model performance and select the best performing model. I will tune the best performing model of the three and finally see how it performs on the test data.

Analysis

Data Exploration

To start off, I need to do some simple data exploration to understand more about the data. For this, let's take a quick look at the data itself:

Table 4: Data summary

Name	employee
Number of rows	1197
Number of columns	12
Column type frequency:	
factor	3
numeric	9
Group variables	None

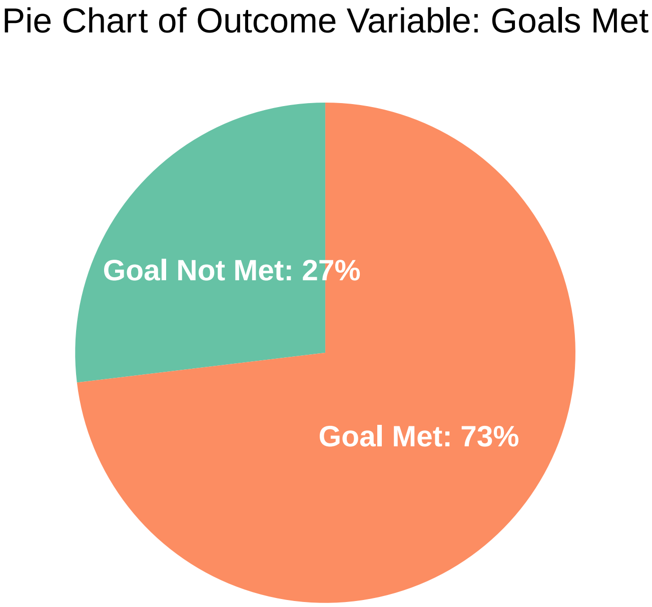
Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
quarter	0	1	FALSE	5	Qua: 360, Qua: 335, Qua: 248, Qua: 210
team	0	1	FALSE	12	2: 109, 8: 109, 1: 105, 4: 105
goal_met	0	1	FALSE	2	1: 875, 0: 322

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
targeted_productivity	0	1	0.73	0.10	0.07	0.70	0.75	0.80	0.80	
smv	0	1	15.06	10.94	2.90	3.94	15.26	24.26	54.56	
wip	0	1	687.23	1514.58	0.00	0.00	586.00	1083.00	23122.00	
over_time	0	1	4567.46	3348.82	0.00	1440.00	3960.00	6960.00	25920.00	
incentive	0	1	38.21	160.18	0.00	0.00	0.00	50.00	3600.00	
idle_time	0	1	0.73	12.71	0.00	0.00	0.00	0.00	300.00	
idle_men	0	1	0.37	3.27	0.00	0.00	0.00	0.00	45.00	
no_of_style_change	0	1	0.15	0.43	0.00	0.00	0.00	0.00	2.00	
no_of_workers	0	1	34.61	22.20	2.00	9.00	34.00	57.00	89.00	

We can also look at the distribution or proportion of the outcome variable `goal_met`, see pie chart below.

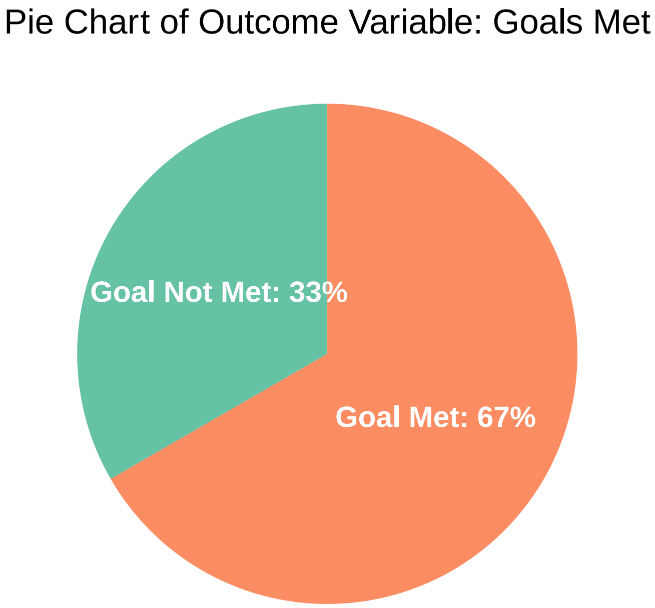


Recipe

```r
rec <- recipe(goal_met ~ ., data = train) %>%
  step_upsample(goal_met, over_ratio = 0.5) %>%
  step_normalize(all_numeric_predictors()) %>%
  step_dummy(all_factor_predictors())
```

For the recipe, I've added a `step_normalize` for all numeric predictors, and a `step_dummy` for the factor predictors.

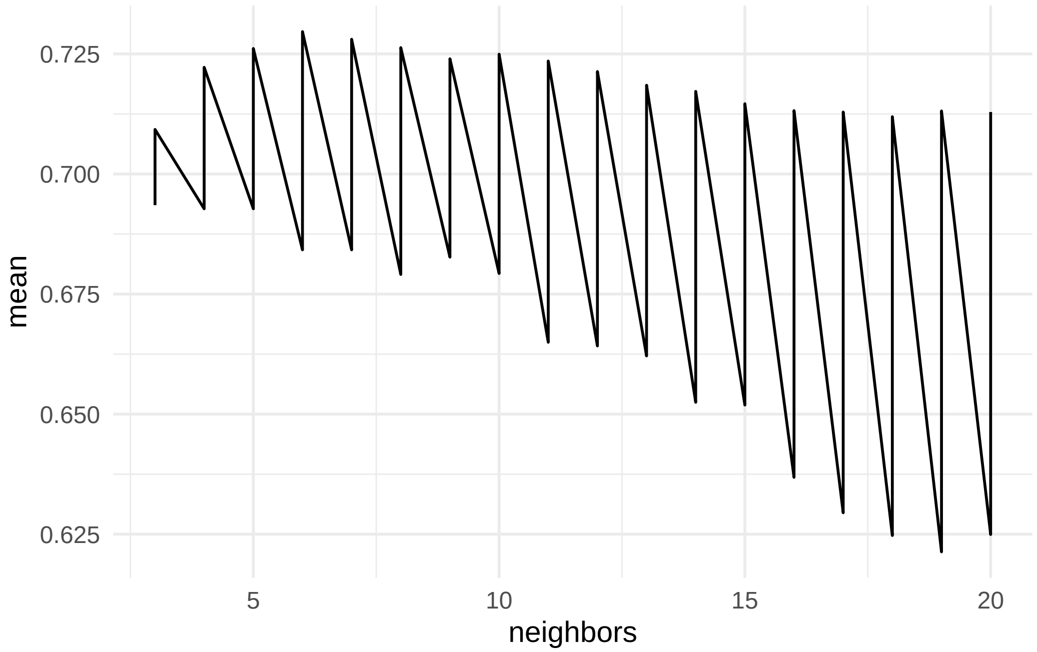
Based outcome variable's imbalance, I have also chosen to upsample the goal not being met, to help with training the model. I've set the ratio to 50%, as is shown below:



Training Models: Naive Bayes, kNN, Random Forest, XGBoost

I created and trained four different machine learning models to compare performances: Naive Bayes, kNN, Random Forest, XGBoost. I only want to tune a limited number of models in order to save computational power, so I will be first training the models without tuning the parameters. Once I can compare performance between the four models, I will tune the best performing model, and then fit it to the test data.

Although, I do not intend to tune the models before I compare them, I did choose to tune the kNN model in order to select the number of neighbors to use. I made this exception to avoid choosing randomly, without any prior reasoning for that value. The results of the tuning process are below.



```
[1] "Optimal number of neighbors for kNN model: 3"
```

Collecting Metrics

After having fit all models to the training data, I can collect the metrics for all four and choose the best performing model.

```
# A tibble: 4 x 3
  Model                 `ROC AUC` `Balanced Accuracy`
  <chr>                     <dbl>               <dbl>
1 Random Forest Model       0.862               0.751
2 XGBoosted Tree Model      0.858               0.751
3 Naive Bayes Model         0.752               0.513
4 kNN Model                 0.709               0.694
```

The performance of the Random Forest model and the XGBoosted Tree model were nearly identical. The Random Forest slightly outperformed the XGBoosted Tree when looking at the ROC AUC, but the XGBoosted Tree slightly outperformed the Random Forest on balanced accuracy. Based on how close these two models performed, I will tune both, and evaluate which performs better after tuning on the resampled data. This estimation will allow me to see which might be better to evaluate performance on the testing data.

Tuning the XGBoosted Tree Model

```
[1] "Optimal mtry value for XGBoost model: 12"
```

```
[1] "Optimal tree depth for XGBoost model: 14"
```

```
[1] "Optimal learn rate for XGBoost model: 0"
```

```
[1] "Optimal loss reduction for XGBoost model: 0"
```

```
[1] "Optimal sample size for XGBoost model: 0.928"
```

```
[1] "Optimal number of trees for XGBoost model: 1858"
```

Tuning the Random Forest Model

```
[1] "Optimal mtry value for Random Forest model: 10"
```

```
[1] "Optimal min_n value for Random Forest model: 2"
```

```
[1] "Optimal number of trees for Random Forest model: 1000"
```

Comparing Tuned Performance

```
# A tibble: 2 x 3
  model               bal_accuracy roc_auc
  <chr>                      <dbl>   <dbl>
1 Random Forest Model        0.782   0.871
2 XGBoosted Model            0.771   0.862
```

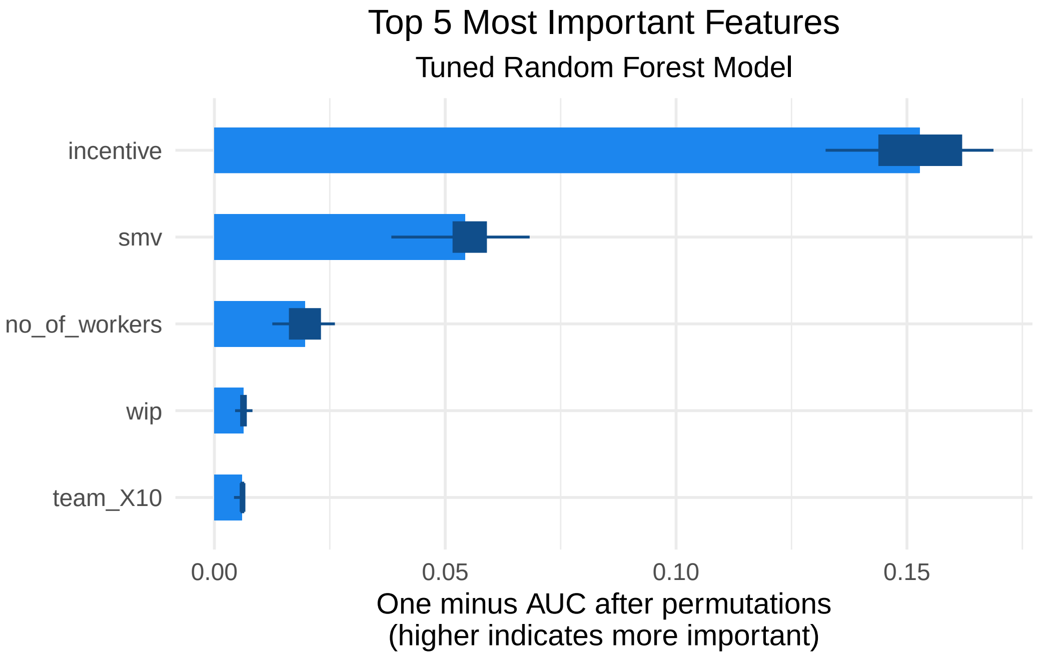
The tuned Random Forest model slightly outperformed the XGBoosted model on the resampled training data. Based on this result, the Random Forest model is the one that will be evaluated on the testing data and examined further.

Evaluation on the Testing Data

```
# A tibble: 2 x 3
  .metric      .estimator .estimate
  <chr>        <chr>          <dbl>
1 bal_accuracy binary         0.906
2 roc_auc      binary         0.945
```

The model's performance on the test data was exceptional, even outperforming the estimated performance on the resampled training data. This is encouraging, as it indicates the model performs well on new data and is not overfit to the training set.

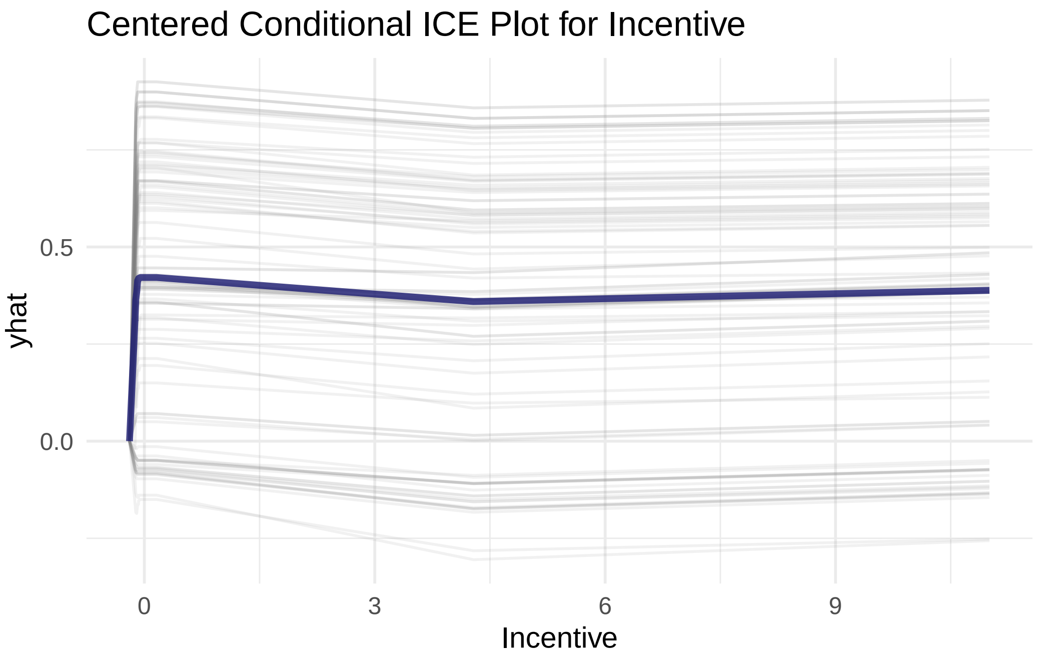
Feature Importance



RQ 1: Does higher monetary incentives predict meeting performance goals?

- *Original Hypothesis:* I hypothesize that incentives will be highly predictive of meeting performance goals.

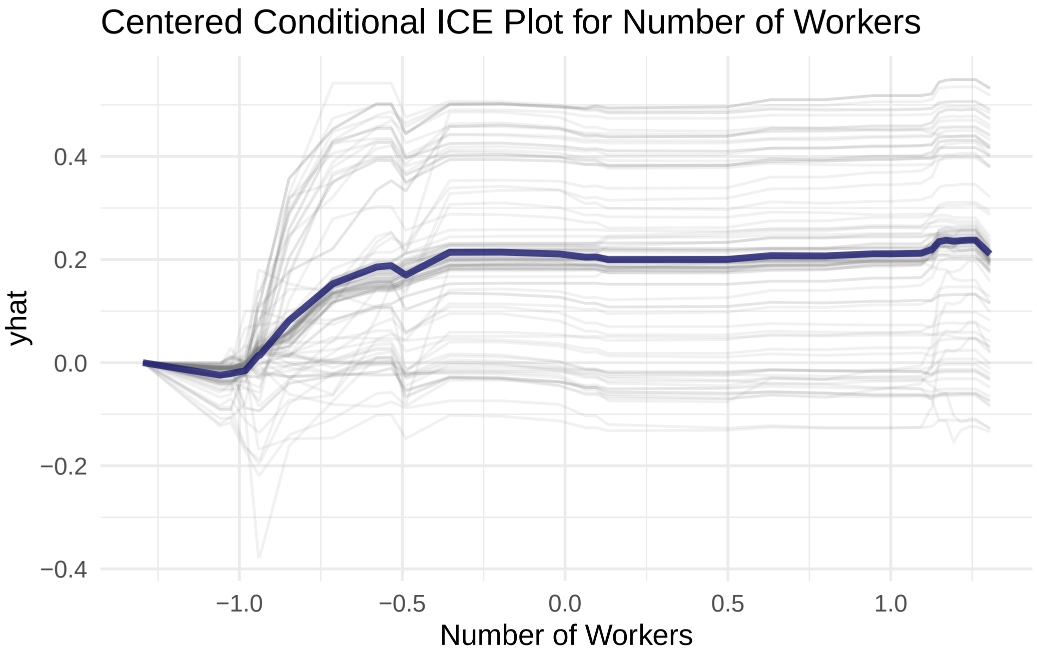
As predicted, incentive was the most important feature in predicting meeting performance goals, Taking a closer look at the feature, reveals that an excessive incentive does not necessarily mean a higher likelihood of meeting the goal for that team, but that incentive equaling or above the average incentive offered (raw mean = 38.21, in BDT), does drastically increase the likelihood of their goals being met:

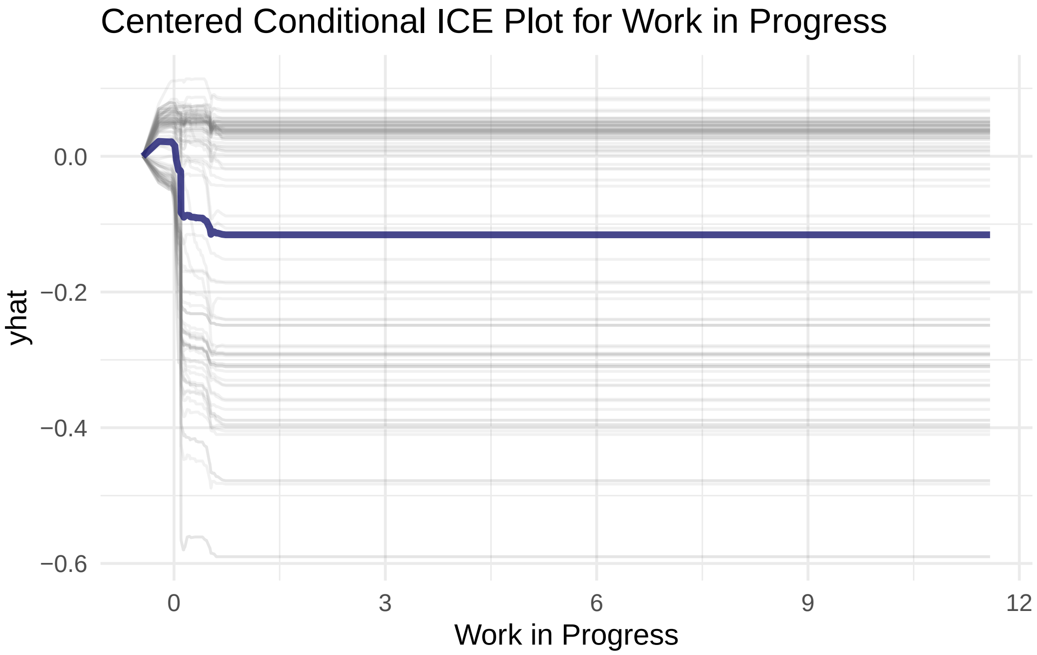


RQ 2: Will the number of workers on a team interact with the number of works in progress to predict performance goals being met?

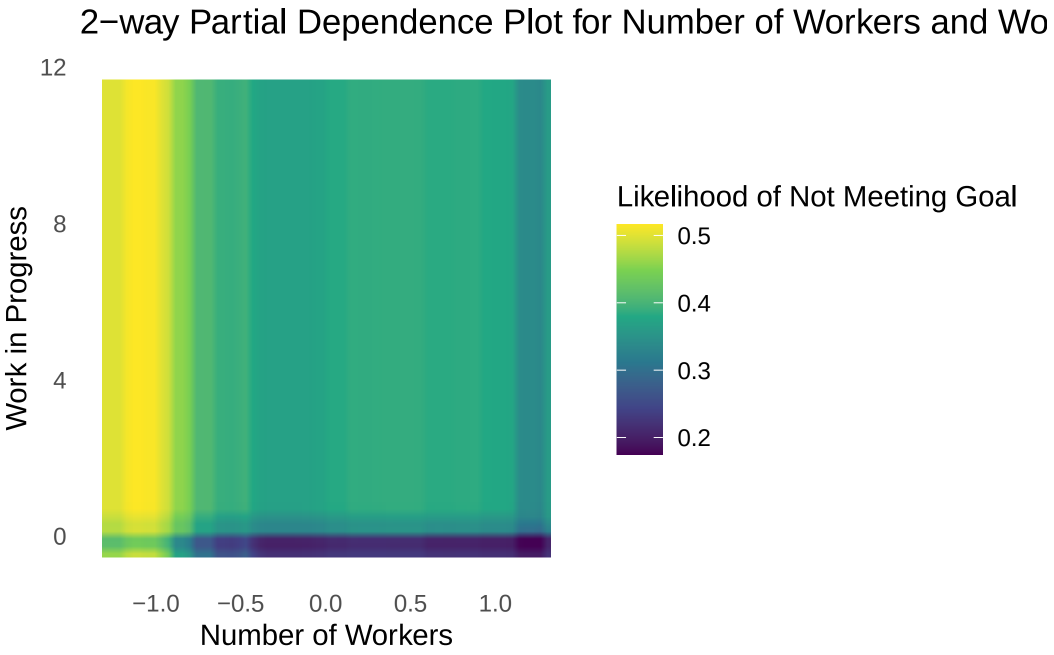
- *Original Hypothesis:* Having a high number of workers alone is likely to be good for meeting performance goals, however, looking at the interaction between the number of workers as well as the number of in progress projects will likely more accurately predict the performance of that team.

As hypothesized, a higher number of workers and the number of works in progress were both among the top 5 features in the model - see ICE plots below for individual features.





I wanted, however, to see if the two variables together seemed to show any kind of dependence on eachother. To explore this, I created the two-way PDP plot below:

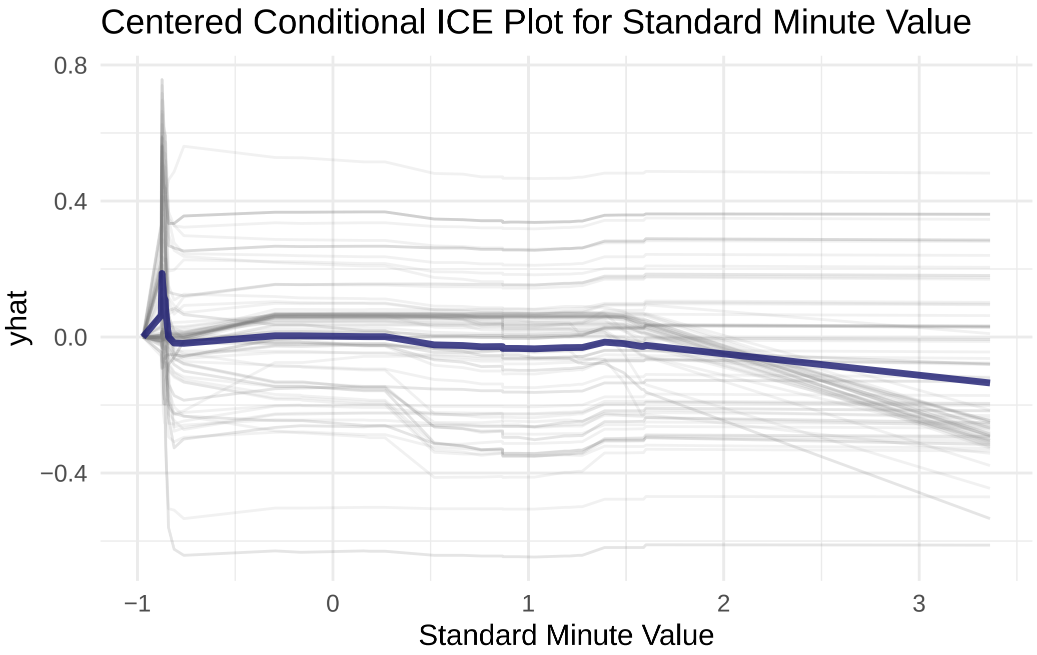


From this plot, it's clear that the hypothesis stating they will be dependent on each other was not supportive. Having fewer workers is predictive of not meeting the team's goals, but this doesn't change depending on the amount of work in progress. I would argue that this is indicative of an appropriate overall balance between the number of workers on a team and the amount of work in progress for that team. If this balance were not in place, a company might be able to locate this issue here.

RQ 3: Will the number of style changes be an important predictor for meeting performance goals?

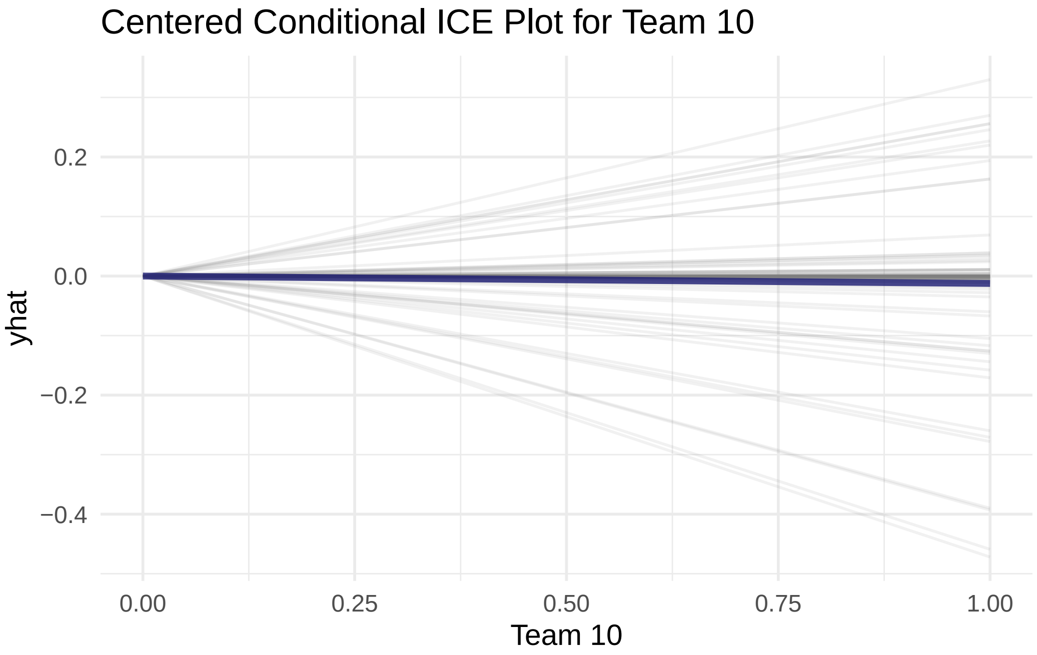
- *Original Hypothesis:* A higher number of style changes likely indicates more complex work that a particular team is doing, and it suggests a greater need for constant adaptation from that team. This constant need for adaptation and constant interruption of their current flow will likely be an important predictor for performance goals being met.

The original hypothesis above was not supported. The number of style changes a team was not among the top 5 (or even top 10) most important features for this model. Instead, the standard minute value (which refers to the allocated time for a task) appeared as the second most important feature for this model.



Lastly, the dummy coded “Team 10” was also an important feature. Based on the PDP below, if this team was the one being evaluated (`team_X10 = 1`), there was a *reduced* likelihood that the goal would be met that day.

This is particularly helpful in identifying a problematic team, or a team that is struggling to maintain high performance.



Section 5 - Conclusion

The tuned random forest model performed best on the training data among all four tested models(balanced accuracy = 0.78, ROC AUC = 0.87). It performed even better on the testing data, indicating that it was not overfit to the training data and is able to perform well on new data (balanced accuracy = 0.91, ROC AUC = 0.94). Incentive, standard minute value, the number of workers, and the amount of work in progress were among the top five most important features in the model. The team #10 was also highly predictive of whether or not the goal would be met.

With the exception of style changes, the variables I hypothesized would rank high in importance did turn out to be important features. The interaction predicted, however, was not supported. Exploring each of the five most important features individually also revealed some important and useful information.

For incentive, if the amount offered was at the mean (raw mean = 38.21, in BDT), we saw a dramatic increase in the likelihood of a goal being met. However, above the mean, there was very little difference seen later. This can be helpful to employers utilizing this model, as it can identify when additional monetary incentive begins to be ineffective.

For the standard minute value, perhaps unsurprisingly, projects which on average took less time, increased the likelihood of meeting the goal, while longer projects decreased the likelihood of meeting the goal. This is particularly helpful for an employer as they can identify the project length that begins to have a negative impact on the team's ability to reach their targeted productivity. Such longer projects should be taken into account both when setting realistic goals and determining how many people a team should include.

The number of workers on a team and the amount of work in progress were also important features in predicting if the goal was met or not. As expected, more workers tended to correspond to a greater likelihood of the goal being met, and the amount of work in progress was negatively predictive of the team's goal being met.

Finally, team 10 was predictive of *not* meeting the goal. Including the team number in the model was helpful as it allowed the model to identify a particularly problematic team that might need extra support or interventions to meet their goals. Had the team been positively predicting of meeting the goal, the employer might look into the team's system to see how to help other teams be as efficient as them.

References:

Abdullah Al Imran, Md Shamsur Rahim, and Tanvir Ahmed (2021) Mining the productivity data of the garment industry. Int. J. Bus. Intell. Data Min. 19, 3 (2021), 319–342. https://doi.org/10.1504/ijbidm.2021.118183

Balla, I., Rahayu, S., & Purnama, J. J. (2021). GARMENT EMPLOYEE PRODUCTIVITY PREDICTION USING RANDOM FOREST. Techno Nusa Mandiri: Journal of Computing and Information Technology, 18(1), 49–54. https://doi.org/10.33480/techno.v18i1.2210

Productivity Prediction of Garment Employees [Dataset]. (2020). UCI Machine Learning Repository. https://doi.org/10.24432/C51S6D.