HarvardX- Choose Your Own Project: Productivity prediction for Garment Industry

Gaurav Shrivastava

1/16/2022

Content:

- Report has 6 sections as follows:-
  - Executive Summary
  - loading Library & data and basic summary statics
  - Data Cleaning
  - Exploratory data analysis
  - Machine Learning Model Building
  - Results
  - Conclusion

Executive Summary

- For capstone project of HarvardX, I have taken garment industry data from UCI machine learning database. The Objective of project is to predict productivity of various departments of garment industry. The dataset is taken from following website for the project :-

"https://archive.ics.uci.edu/ml/machine-learning-databases/00597/garments_worker_productivity.csv"

- Dataset has following features/parameters
  - date : Date in MM-DD-YYYY format
  - day : Day of the Week
  - quarter : A portion of the month. A month was divided into four quarters
  - department : Associated department with the instance
  - team_no : 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, it 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

- Approach used for building machine learning model included dividing dataset into training and test set. Then, building prediction model on training set. For validation, the model built on training set is used to predict productivity on test set. 80% observation is used for training of the model and 20% of observation from dataset is used for testing.
- To check the accuracy of model root mean square error(RMSE) is estimated and model having lowest RMSE is selected as final machine learning model for the project.
- Garment industry is a highly labour-intensive industry with lots of manual processes. Satisfying the huge global demand for garment products is mostly dependent on the production and delivery performance of the employees in the garment manufacturing companies. So, it is highly desirable among the decision makers in the garments industry to track, analyze and predict the productivity performance of the working teams in their factories.
- To achieve above business purpose, I classified productivity into 4 categories for proactive decision making from top management. In this project, those categories are ranked as 1,2,3,4 where 4 represents highest productivity group and 1 represents the lowest productivity group.
- In following sections, I have explained in details various steps taken to build the machine learning algorithm on dataset.

Loading Library & data and basic summary statics

- Following libraries are installed and loaded before start building model programs: library(tidyverse) library(caret) library(ggplot2) library(dslabs) library(ggrepel) library(dplyr) library(lubridate) library(HistData) library(purrr) library(pdftools) library(matrixStats) library(genefilter) library(randomForest) library(readxl)
- In this step, I downloaded the dataset file using R codes and understood basic structure and properties of the database. Following code is used to download the file from internet
- Dataset has 15 features and 1197 observations. Following is detail about dataset:-

```
## [1] 1197   15
```

```
## spec_tbl_df [1,197 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ date                 : chr [1:1197] "1/1/2015" "1/1/2015" "1/1/2015" "1/1/2015" ...
##  $ quarter              : chr [1:1197] "Quarter1" "Quarter1" "Quarter1" "Quarter1" ...
##  $ department           : chr [1:1197] "sweing" "finishing" "sweing" "sweing" ...
##  $ day                  : chr [1:1197] "Thursday" "Thursday" "Thursday" "Thursday" ...
##  $ team                 : num [1:1197] 8 1 11 12 6 7 2 3 2 1 ...
##  $ targeted_productivity: num [1:1197] 0.8 0.75 0.8 0.8 0.8 0.8 0.75 0.75 0.75 0.75 ...
##  $ smv                  : num [1:1197] 26.16 3.94 11.41 11.41 25.9 ...
##  $ wip                  : num [1:1197] 1108 NA 968 968 1170 ...
##  $ over_time            : num [1:1197] 7080 960 3660 3660 1920 6720 960 6900 6000 6900 ...
##  $ incentive            : num [1:1197] 98 0 50 50 50 38 0 45 34 45 ...
##  $ idle_time            : num [1:1197] 0 0 0 0 0 0 0 0 0 0 ...
##  $ idle_men             : num [1:1197] 0 0 0 0 0 0 0 0 0 0 ...
##  $ no_of_style_change   : num [1:1197] 0 0 0 0 0 0 0 0 0 0 ...
##  $ no_of_workers        : num [1:1197] 59 8 30.5 30.5 56 56 8 57.5 55 57.5 ...
##  $ actual_productivity  : num [1:1197] 0.941 0.886 0.801 0.801 0.8 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   date = col_character(),
```

```
##   ..   quarter = col_character(),
##   ..   department = col_character(),
##   ..   day = col_character(),
##   ..   team = col_double(),
##   ..   targeted_productivity = col_double(),
##   ..   smv = col_double(),
##   ..   wip = col_double(),
##   ..   over_time = col_double(),
##   ..   incentive = col_double(),
##   ..   idle_time = col_double(),
##   ..   idle_men = col_double(),
##   ..   no_of_style_change = col_double(),
##   ..   no_of_workers = col_double(),
##   ..   actual_productivity = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

- First six observation of dataset is as follows. It starts with date and ends with final column which is actual productivity:-

```
## # A tibble: 6 x 15
##   date     quarter department day    team targeted_produc~   smv   wip over_time
##   <chr>    <chr>   <chr>      <chr> <dbl>            <dbl> <dbl> <dbl>     <dbl>
## 1 1/1/2015 Quarte~ sweing     Thur~     8             0.8  26.2   1108      7080
## 2 1/1/2015 Quarte~ finishing  Thur~     1             0.75  3.94    NA       960
## 3 1/1/2015 Quarte~ sweing     Thur~    11             0.8  11.4    968      3660
## 4 1/1/2015 Quarte~ sweing     Thur~    12             0.8  11.4    968      3660
## 5 1/1/2015 Quarte~ sweing     Thur~     6             0.8  25.9   1170      1920
## 6 1/1/2015 Quarte~ sweing     Thur~     7             0.8  25.9    984      6720
## # ... with 6 more variables: incentive <dbl>, idle_time <dbl>, idle_men <dbl>,
## #   no_of_style_change <dbl>, no_of_workers <dbl>, actual_productivity <dbl>
```

Data Cleaning

- As next step, I did some data wrangling to change data as per the requirement for modeling. In this step, I will Change data type of features from character to factor for quarter,department,day and team features using following code:-

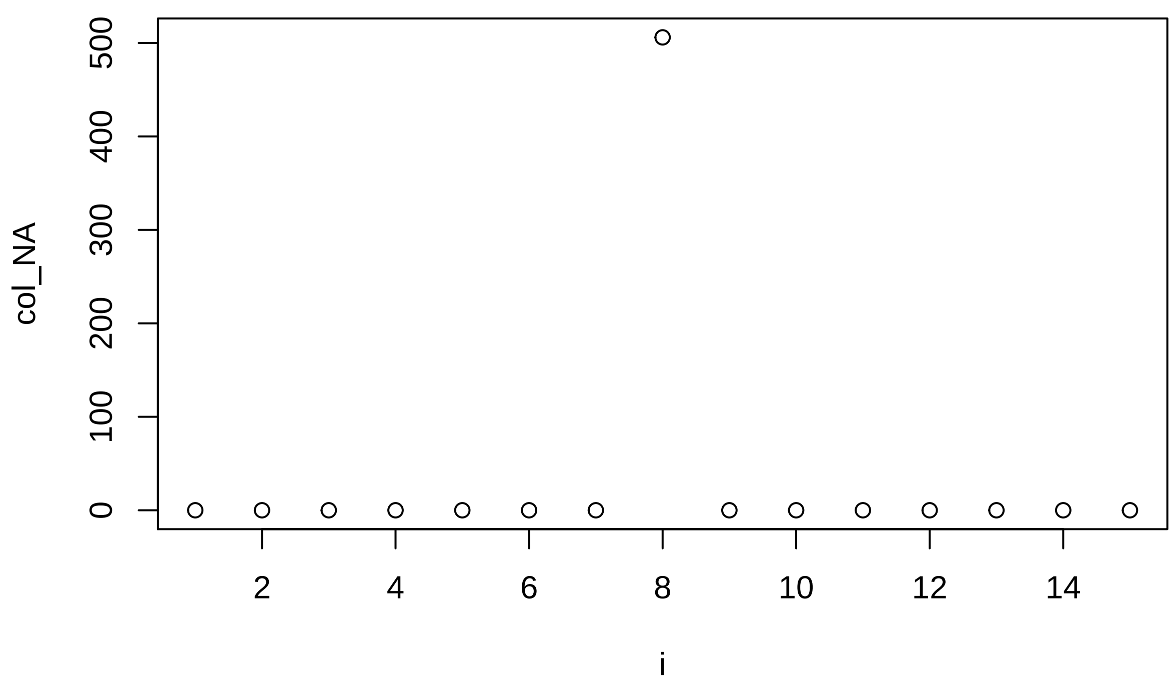
```
dat<-dat%>% mutate(quarter=as.factor(quarter),
                   department=as.factor(department),day=as.factor(day),
                   team=as.factor(team))
```

- Next, I identified columns where NA is used and replaced it with appropriate numbers. For identifying columns, I used following code

```
i<-1:15
col_NA<-sapply(i,function(l)
  dat%>% filter(is.na(dat[,l]))%>% summarise(n=n())
)
```

*Following chart showed that only feature number 8 has NA values. Name of that column is “wip”. I replaced NA values with 0 using following code

```r
plot(i,col_NA)
```




```r
names(dat[col_NA>0])
```

```
## [1] "wip"
```

```r
dat<-dat%>%mutate(wip=ifelse(is.na(wip),0,wip))
```

- First six observations of datasets after data cleaning is follows:-

```
## # A tibble: 6 x 15
##   date     quarter department day   team  targeted_produc~   smv   wip over_time
##   <chr>    <fct>   <fct>      <fct> <fct>            <dbl> <dbl> <dbl>     <dbl>
## 1 1/1/2015 Quarte~ sweing     Thur~ 8                 0.8  26.2   1108      7080
## 2 1/1/2015 Quarte~ finishing  Thur~ 1                 0.75  3.94     0       960
## 3 1/1/2015 Quarte~ sweing     Thur~ 11                0.8  11.4    968      3660
## 4 1/1/2015 Quarte~ sweing     Thur~ 12                0.8  11.4    968      3660
## 5 1/1/2015 Quarte~ sweing     Thur~ 6                 0.8  25.9   1170      1920
## 6 1/1/2015 Quarte~ sweing     Thur~ 7                 0.8  25.9    984      6720
## # ... with 6 more variables: incentive <dbl>, idle_time <dbl>, idle_men <dbl>,
## #   no_of_style_change <dbl>, no_of_workers <dbl>, actual_productivity <dbl>
```

#Exploratory data Analysis:-

- In this step, I went more deeper in the data by understanding summary statistics and trends between various features. Below is summary statistics of the dataset using following code:-

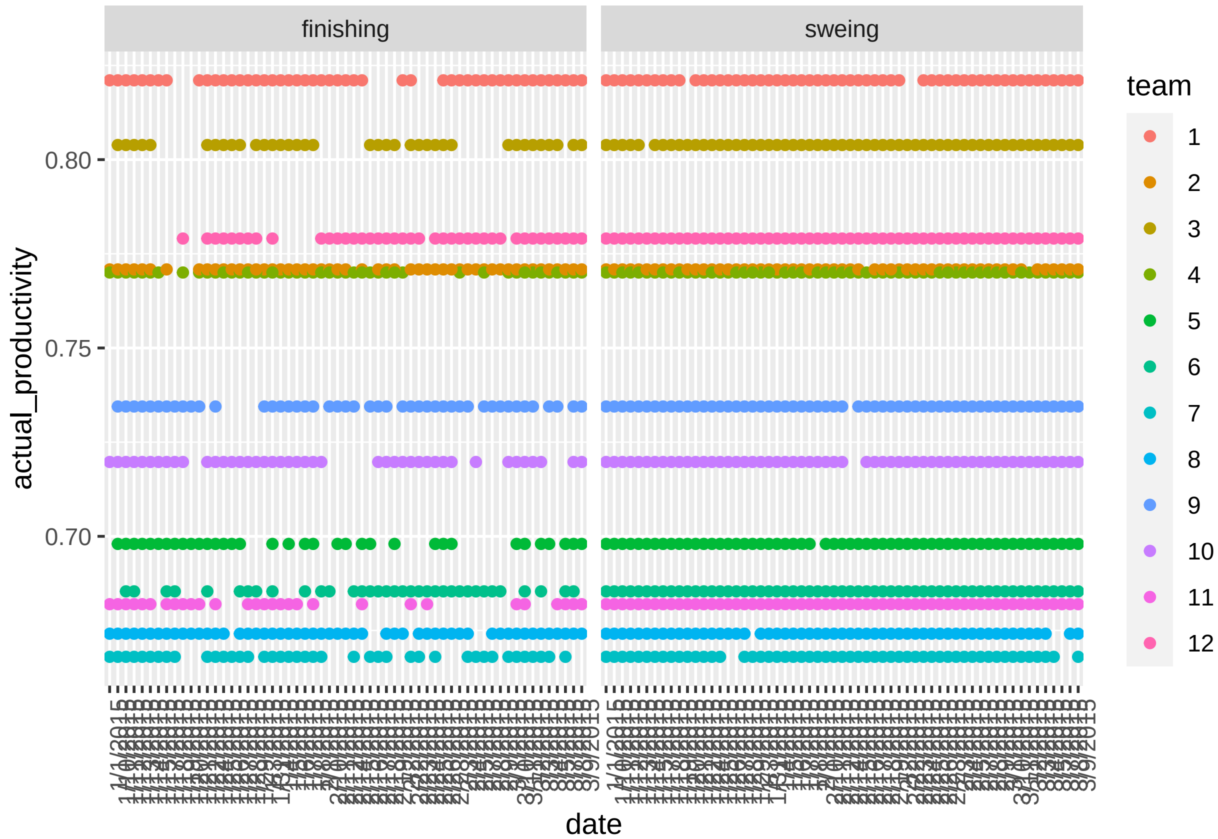
```
summary(dat)
```

```
##      date              quarter       department         day
##  Length:1197        Quarter1:360   finishing:506   Monday   :199
##  Class :character   Quarter2:335   sweing   :691   Saturday :187
##  Mode  :character   Quarter3:210                   Sunday   :203
##                     Quarter4:248                   Thursday :199
##                     Quarter5: 44                   Tuesday  :201
##                                                    Wednesday:208
##
##       team     targeted_productivity      smv             wip
##  2      :109   Min.   :0.0700        Min.   : 2.90   Min.   :    0.0
##  8      :109   1st Qu.:0.7000        1st Qu.: 3.94   1st Qu.:    0.0
##  1      :105   Median :0.7500        Median :15.26   Median :  586.0
##  4      :105   Mean   :0.7296        Mean   :15.06   Mean   :  687.2
##  9      :104   3rd Qu.:0.8000        3rd Qu.:24.26   3rd Qu.: 1083.0
##  10     :100   Max.   :0.8000        Max.   :54.56   Max.   :23122.0
##  (Other):565
##    over_time       incentive         idle_time          idle_men
##  Min.   :    0   Min.   :   0.00   Min.   :  0.0000   Min.   : 0.0000
##  1st Qu.: 1440   1st Qu.:   0.00   1st Qu.:  0.0000   1st Qu.: 0.0000
##  Median : 3960   Median :   0.00   Median :  0.0000   Median : 0.0000
##  Mean   : 4567   Mean   :  38.21   Mean   :  0.7302   Mean   : 0.3693
##  3rd Qu.: 6960   3rd Qu.:  50.00   3rd Qu.:  0.0000   3rd Qu.: 0.0000
##  Max.   :25920   Max.   :3600.00   Max.   :300.0000   Max.   :45.0000
##
##  no_of_style_change no_of_workers   actual_productivity
##  Min.   :0.0000     Min.   : 2.00   Min.   :0.2337
##  1st Qu.:0.0000     1st Qu.: 9.00   1st Qu.:0.6503
##  Median :0.0000     Median :34.00   Median :0.7733
##  Mean   :0.1504     Mean   :34.61   Mean   :0.7351
##  3rd Qu.:0.0000     3rd Qu.:57.00   3rd Qu.:0.8503
##  Max.   :2.0000     Max.   :89.00   Max.   :1.1204
##
```

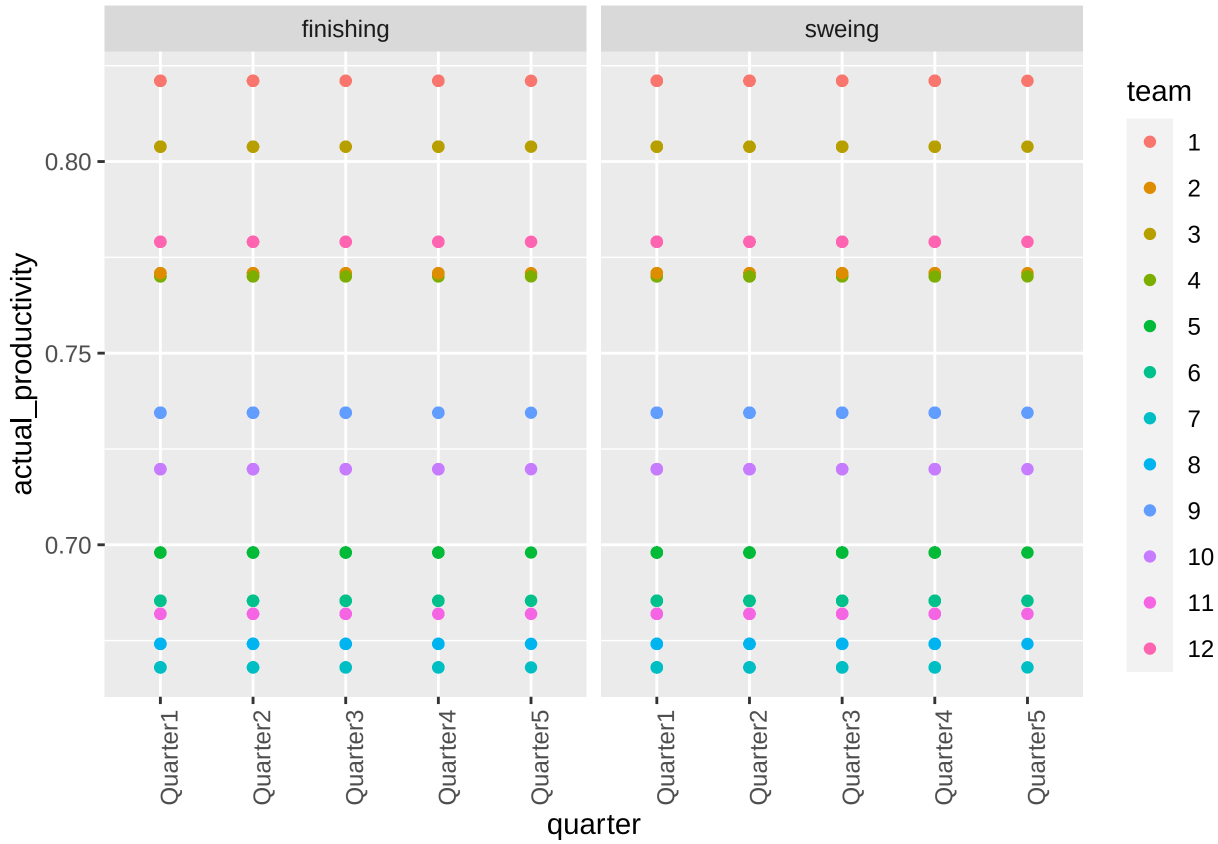
- Distinct number of elements in different features of dataset is as follows. Only dataset whose class is factor.

```
## # A tibble: 1 x 5
##   n_date n_quarter n_department n_day n_team
##    <int>     <int>        <int> <int>  <int>
## 1     59         5            2     6     12
```

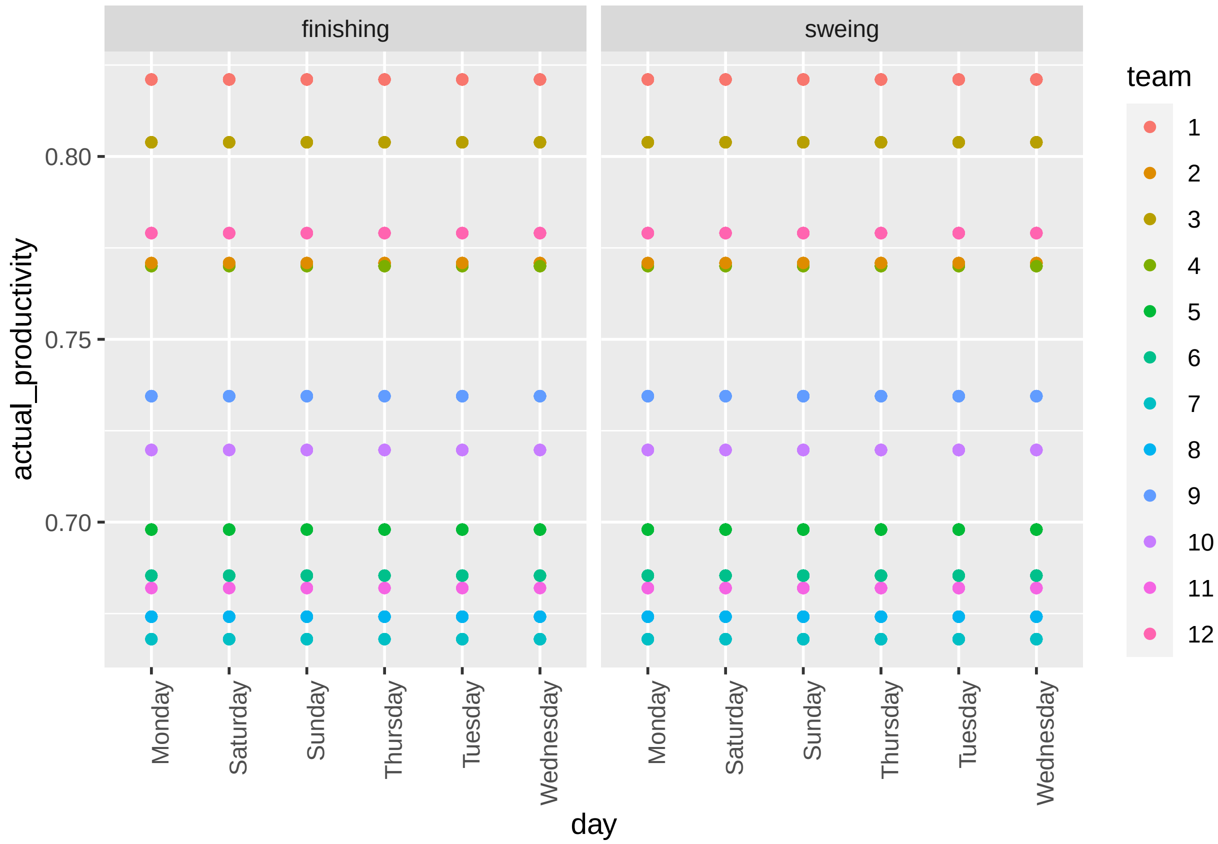
- Further, I analyzed trends of actual productivity with respect to various parameters:-
  - Department/team- wise Average actual productivity vs date chart:- Average actual productivity is same across various dates of different functions. Some team has higher productivity and some team has lower productivity. We can say teamwise average productivity per day is constant



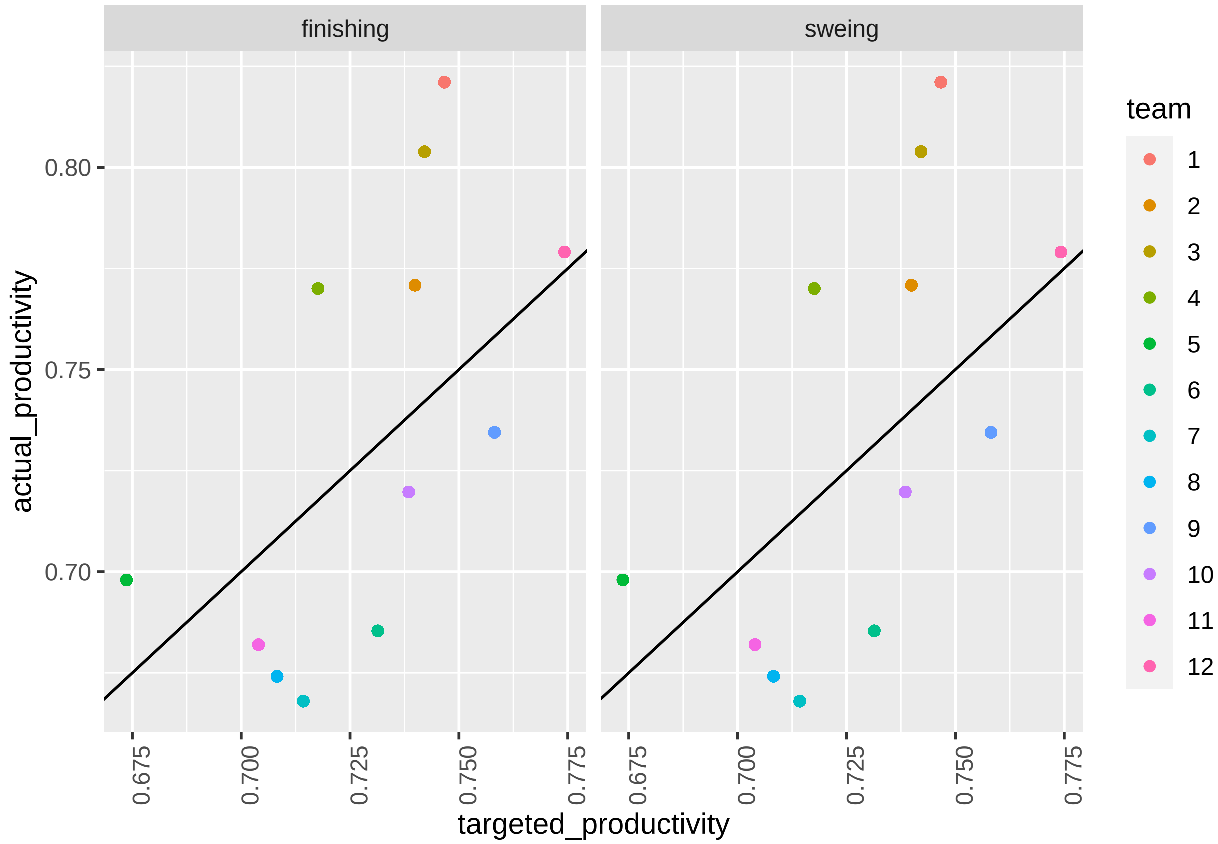
- Department/team- wise Average actual productivity vs quarter chart:- Average actual productivity is same across various quarters of month for different functions. Some team has higher productivity and some team has lower productivity. We can say teamwise average productivity per quarter is constant



- Department/team- wise Average actual productivity vs day chart:- Average actual productivity is same across various days for different functions. Some team has higher productivity and some team has lower productivity. We can say teamwise average productivity per day is constant

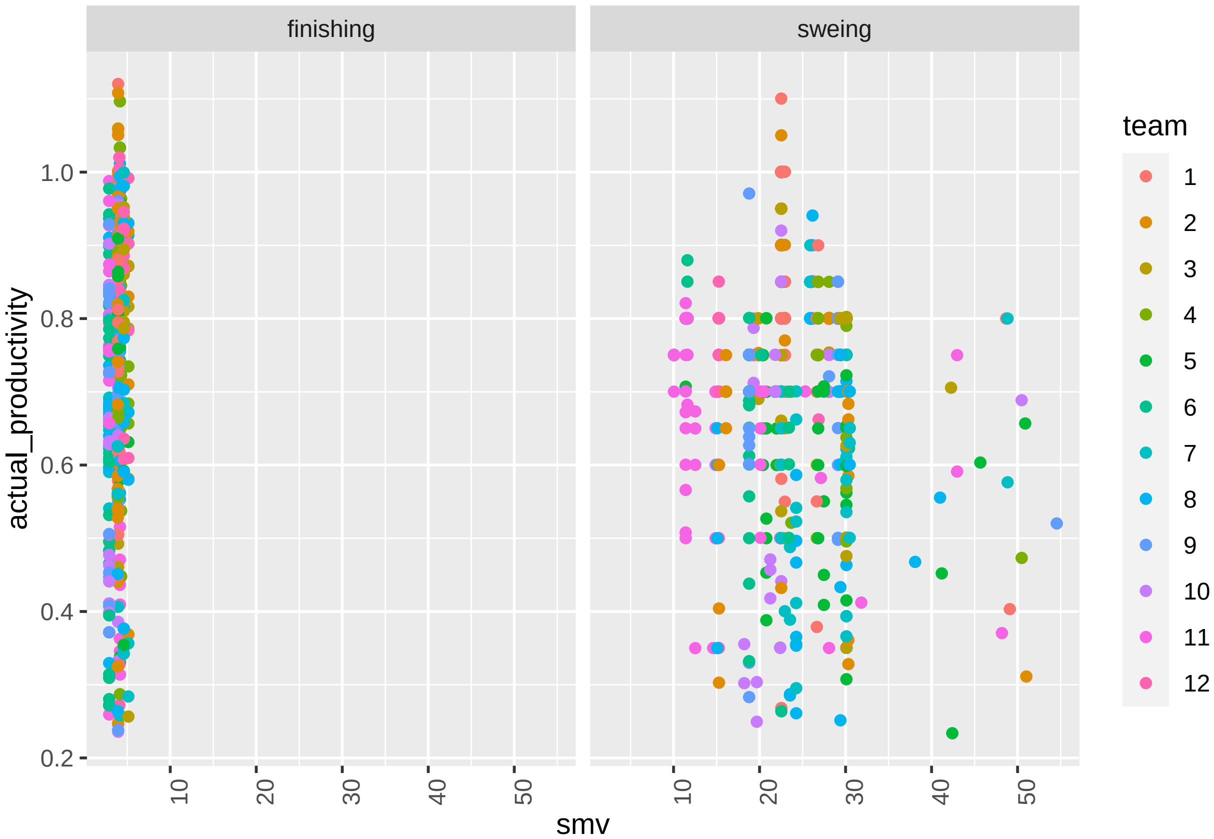


- Department/team- wise Average actual productivity vs average targeted_productivity chart:- Average actual productivity has increasing trend with average targeted productivity for different functions.



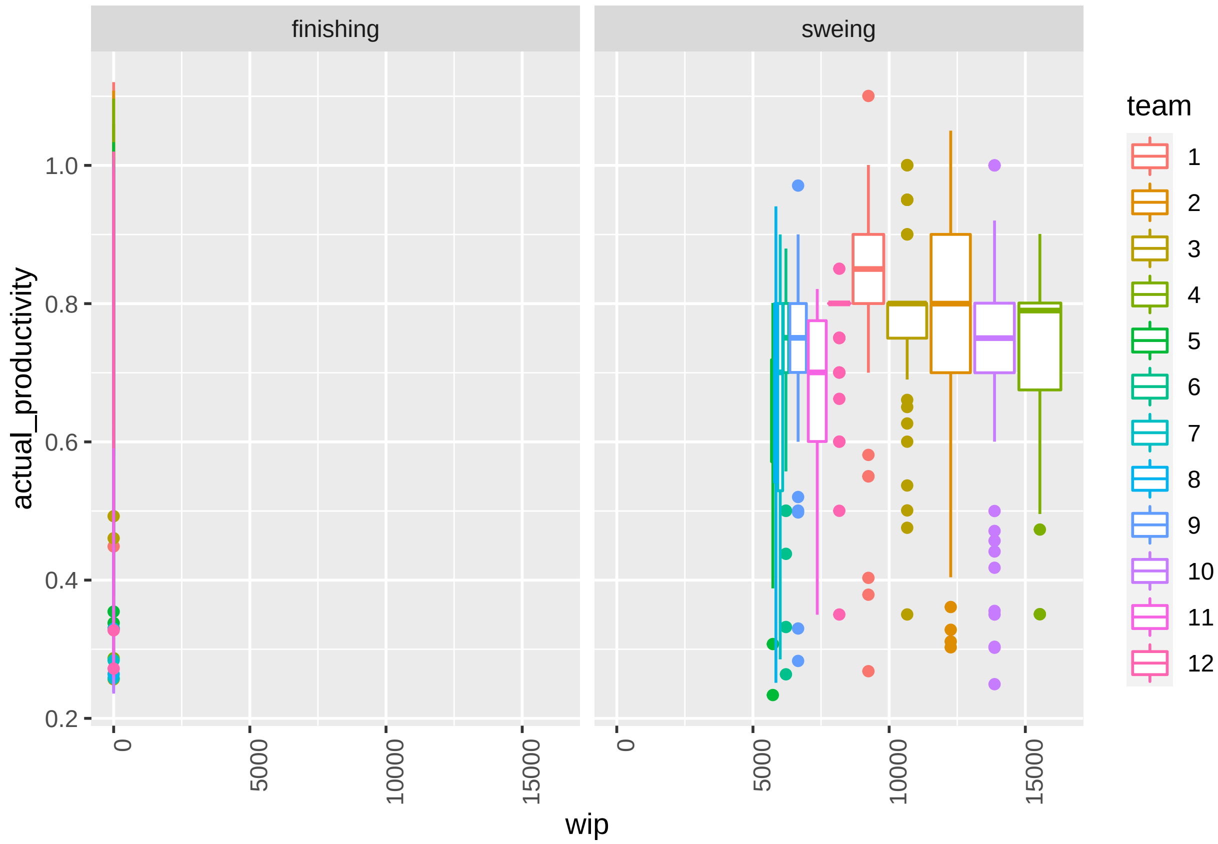
+Department wise actual productivity vs smv chart:-

```
Actual productivity has no trend with smv for different functions.
```



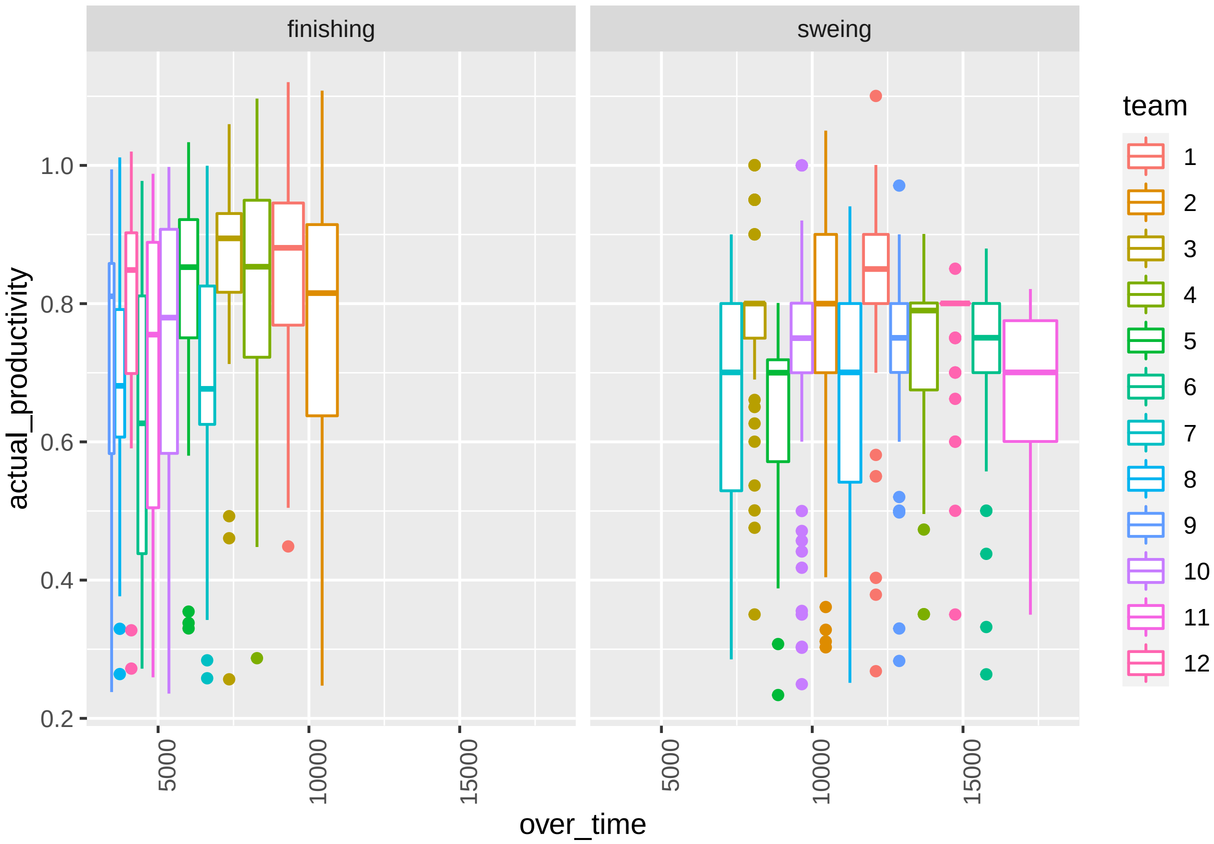
- Department wise actual productivity vs wip chart:-

  Actual productivity has no significant trend with average wip for different functions.



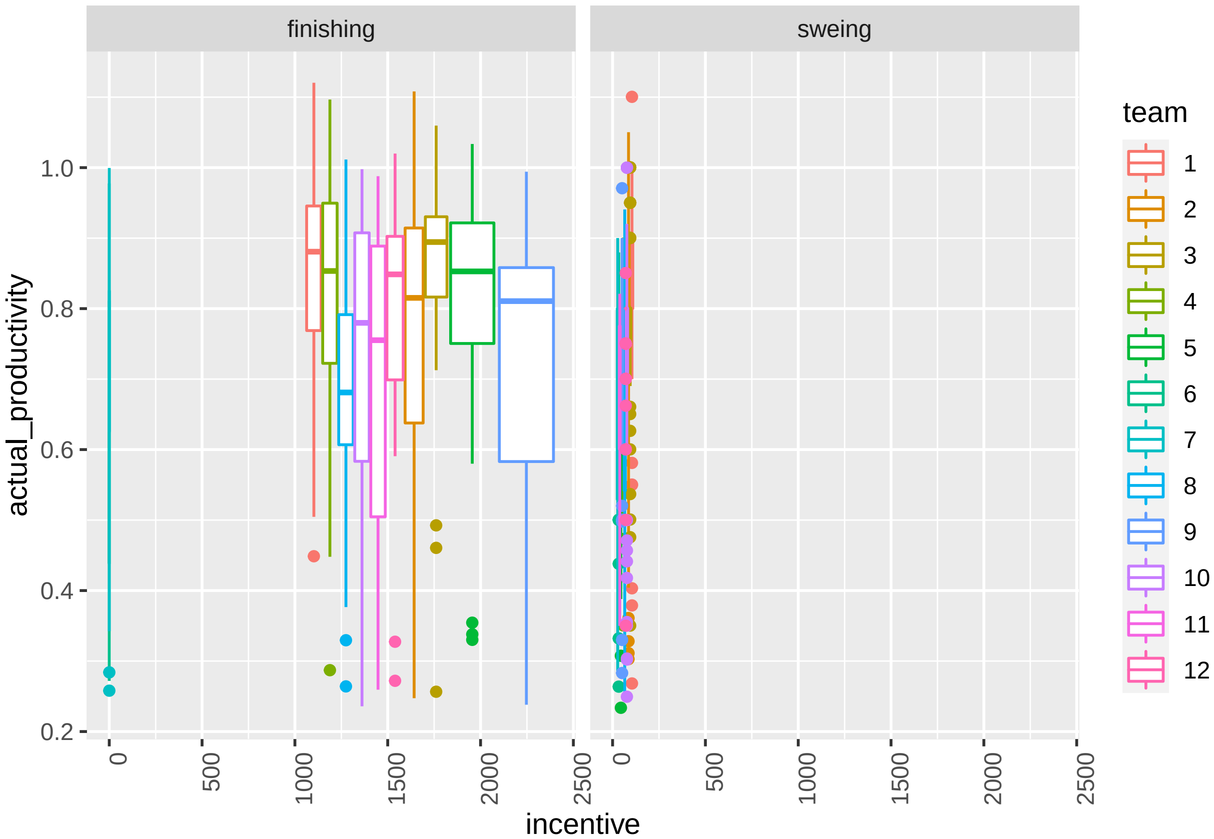
- Department wise actual productivity vs over_time chart:-

  Actual productivity has no trend with average over_time for different functions.

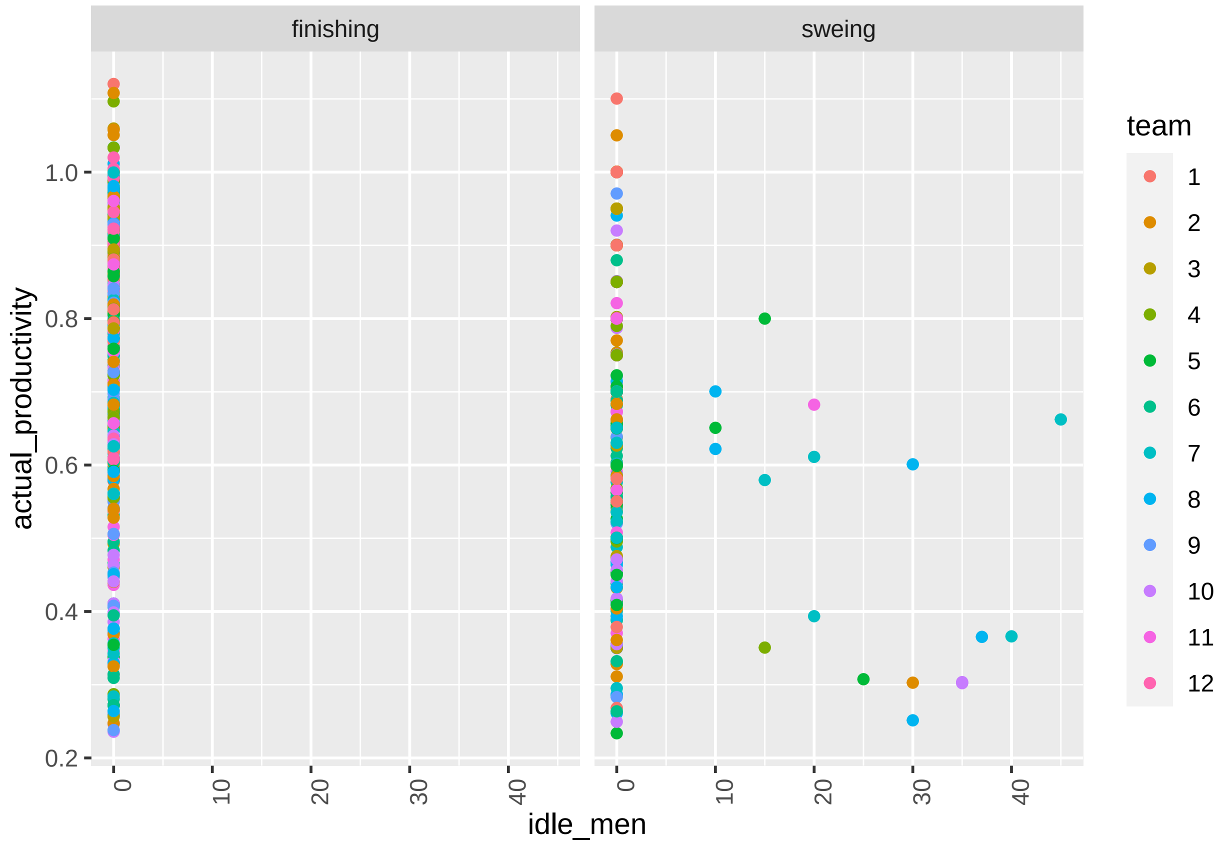


- Department wise actual productivity vs incentive chart:-

  Actual productivity has no trend with incentive for different functions.

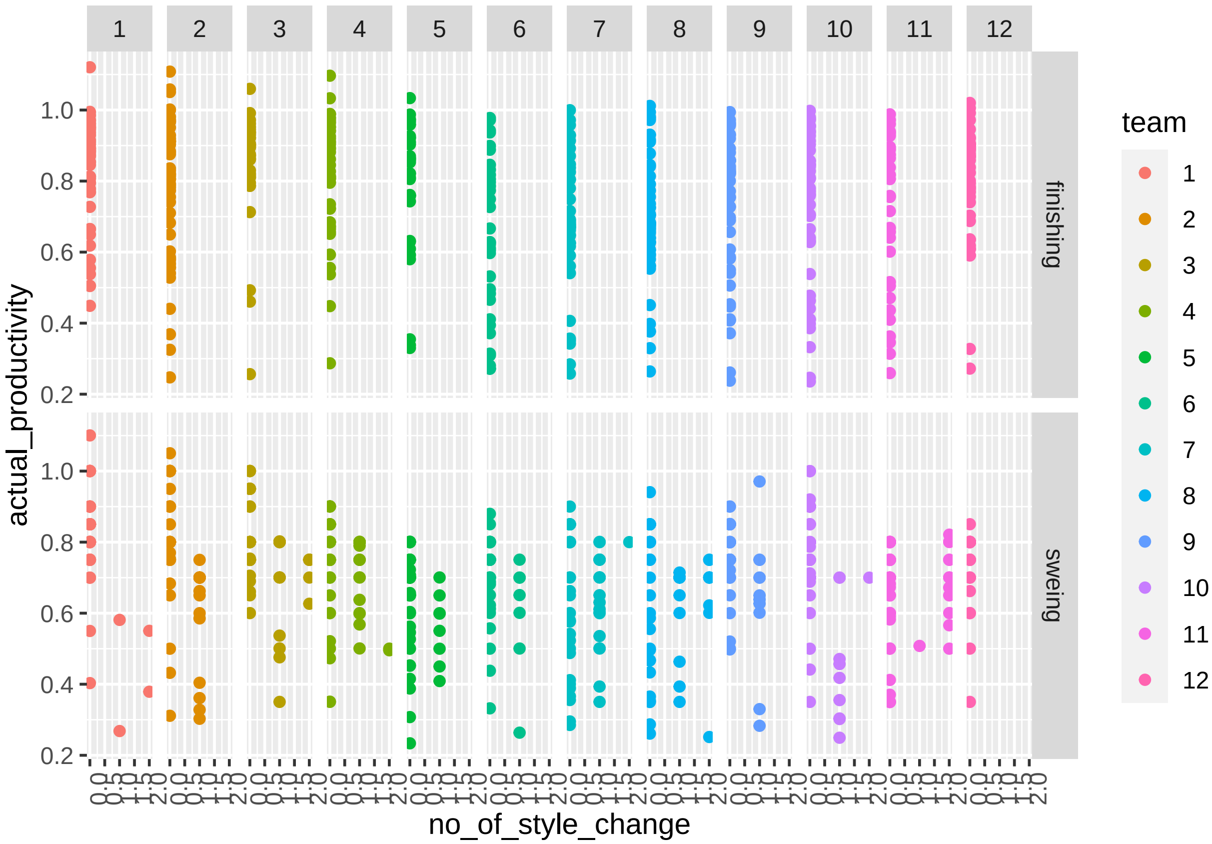


- Department wise actual productivity vs idle_men chart:-
  Actual productivity has no trend with idle_men for different functions.



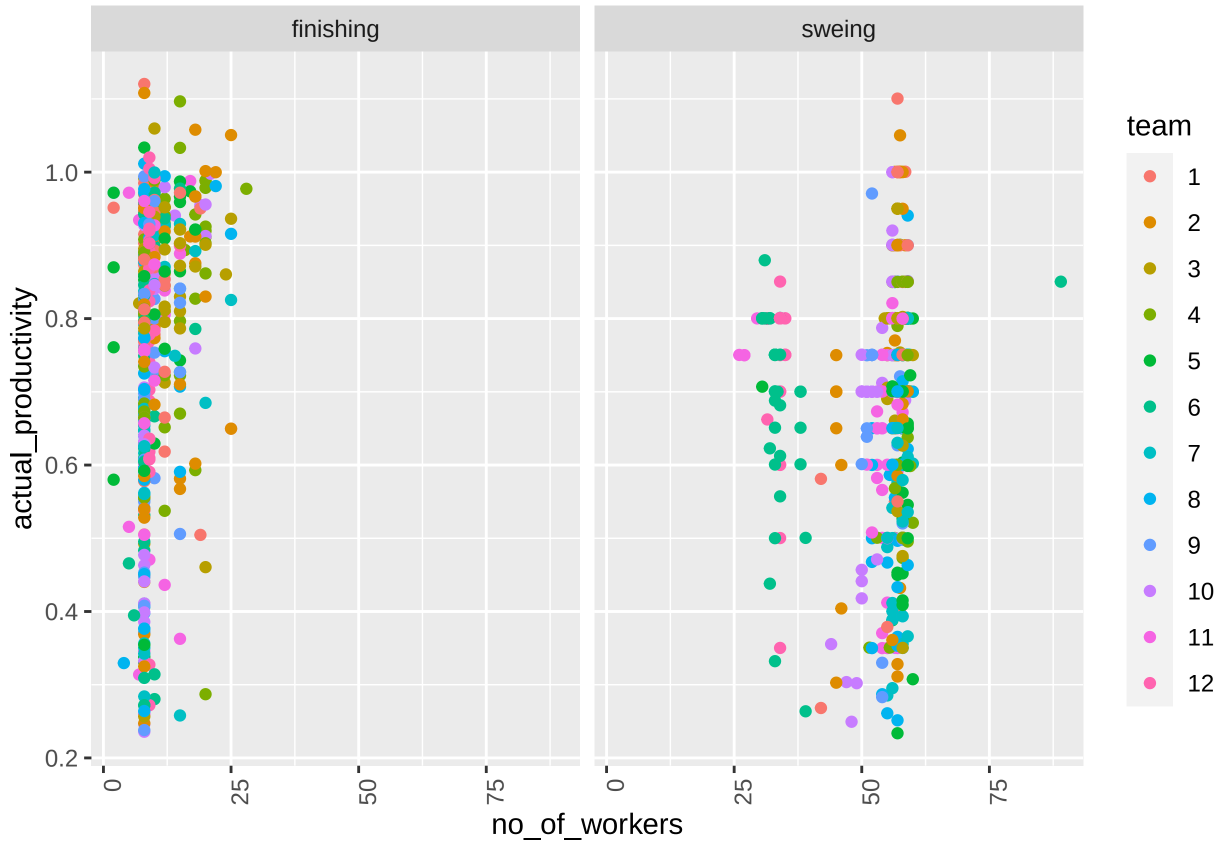
- Department wise actual productivity vs no_of_style_change chart:-

  Actual productivity has no trend with finishing but decreases for sweing department



- Department wise actual productivity vs no_of_workers chart:-

  Actual productivity has no visible trend with no_of_workers for different functions



Machine Learning Model Building

- Before creating machine learning algorithm, we modify the dataset by classifying actual productivity in 4 classes from 1 to 4 as follows:-
  - 4 means actual productivity 0.8,
  - 3 means productivity is between 0.6 and 0.8,
  - 2 means productivity is between 0.4 and 0.6 and
  - 1 means productivity is less than 0.4
- Removing date and actual productivity feature for model building using following code

```
rev_dat<-dat%>%mutate(rev_act_prod=ifelse(actual_productivity>0.8,4,
                                   ifelse(actual_productivity>0.6,3,
                                   ifelse(actual_productivity>0.4,2,1))))%>%
  select(-actual_productivity,-date)
```

```
+ Top six rows of revised dataset is as follows:-
```

```
## # A tibble: 6 x 14
##   quarter  department day      team  targeted_productivity   smv   wip over_time
##   <fct>    <fct>      <fct>    <fct>                 <dbl> <dbl> <dbl>     <dbl>
## 1 Quarter1 sweing     Thursday 8                      0.8   26.2   1108      7080
## 2 Quarter1 finishing  Thursday 1                      0.75   3.94     0       960
```

```
## 3 Quarter1 sweing     Thursday 11                          0.8  11.4    968      3660
## 4 Quarter1 sweing     Thursday 12                          0.8  11.4    968      3660
## 5 Quarter1 sweing     Thursday 6                           0.8  25.9   1170      1920
## 6 Quarter1 sweing     Thursday 7                           0.8  25.9    984      6720
## # ... with 6 more variables: incentive <dbl>, idle_time <dbl>, idle_men <dbl>,
## #   no_of_style_change <dbl>, no_of_workers <dbl>, rev_act_prod <dbl>
```

- Creating training and test datasets for building machine learning algorithm. 80% observation is for training of the model and 20% for testing. I used following code for data partition.

```
set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(y=rev_dat$rev_act_prod, times = 1, p = 0.2, list = FALSE)
train_set <- rev_dat[-test_index,]
test_set <- rev_dat[test_index,]
```

- Top six rows of training dataset is follows:-

```
## # A tibble: 6 x 14
##   quarter  department day      team  targeted_productivity   smv   wip over_time
##   <fct>    <fct>      <fct>    <fct>                 <dbl> <dbl> <dbl>     <dbl>
## 1 Quarter1 sweing     Thursday 8                      0.8  26.2   1108      7080
## 2 Quarter1 finishing  Thursday 1                      0.75  3.94     0       960
## 3 Quarter1 sweing     Thursday 11                     0.8  11.4    968      3660
## 4 Quarter1 sweing     Thursday 12                     0.8  11.4    968      3660
## 5 Quarter1 sweing     Thursday 6                      0.8  25.9   1170      1920
## 6 Quarter1 sweing     Thursday 7                      0.8  25.9    984      6720
## # ... with 6 more variables: incentive <dbl>, idle_time <dbl>, idle_men <dbl>,
## #   no_of_style_change <dbl>, no_of_workers <dbl>, rev_act_prod <dbl>
```

- Top six rows of test dataset is follows:-

```
## # A tibble: 6 x 14
##   quarter  department day      team  targeted_productivity   smv   wip over_time
##   <fct>    <fct>      <fct>    <fct>                 <dbl> <dbl> <dbl>     <dbl>
## 1 Quarter1 sweing     Thursday 5                      0.8  11.4    668      3660
## 2 Quarter1 finishing  Thursday 8                      0.75  2.9      0       960
## 3 Quarter1 sweing     Saturday 10                     0.75 19.3    610      6480
## 4 Quarter1 finishing  Sunday   3                      0.75  4.15     0      1560
## 5 Quarter1 finishing  Sunday   1                      0.8   3.94     0       960
## 6 Quarter1 finishing  Monday   4                      0.8   3.94     0      3840
## # ... with 6 more variables: incentive <dbl>, idle_time <dbl>, idle_men <dbl>,
## #   no_of_style_change <dbl>, no_of_workers <dbl>, rev_act_prod <dbl>
```

- Defining RMSE for model testing using following code:-

```
RMSE <- function(true_ratings, predicted_ratings){
  sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

- Linear regression model: - used following code

```r
fit_lm <- train(rev_act_prod ~ ., method="lm",data = train_set)
```

```
+ summary of linear regression model is as follows:-
```

```r
summary(fit_lm)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6024 -0.3474  0.1127  0.5171  2.2783
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             7.224e-01  2.358e-01   3.064 0.002248 **
## quarterQuarter2         1.326e-02  6.613e-02   0.200 0.841164
## quarterQuarter3        -6.589e-02  7.610e-02  -0.866 0.386748
## quarterQuarter4        -5.200e-02  7.455e-02  -0.698 0.485589
## quarterQuarter5         3.600e-01  1.481e-01   2.431 0.015228 *
## departmentsweing       -5.935e-02  2.060e-01  -0.288 0.773377
## daySaturday             5.681e-02  9.148e-02   0.621 0.534741
## daySunday              -1.197e-02  8.749e-02  -0.137 0.891177
## dayThursday            -2.997e-02  8.837e-02  -0.339 0.734588
## dayTuesday              4.208e-02  8.744e-02   0.481 0.630515
## dayWednesday            2.457e-02  8.614e-02   0.285 0.775503
## team2                  -3.089e-01  1.178e-01  -2.623 0.008863 **
## team3                  -1.577e-02  1.241e-01  -0.127 0.898859
## team4                  -7.863e-02  1.196e-01  -0.657 0.511178
## team5                  -2.835e-01  1.244e-01  -2.279 0.022916 *
## team6                  -4.250e-01  1.393e-01  -3.051 0.002345 **
## team7                  -4.442e-01  1.226e-01  -3.624 0.000306 ***
## team8                  -4.252e-01  1.182e-01  -3.597 0.000339 ***
## team9                  -4.397e-01  1.178e-01  -3.732 0.000202 ***
## team10                 -4.806e-01  1.206e-01  -3.986 7.25e-05 ***
## team11                 -6.558e-01  1.327e-01  -4.944 9.10e-07 ***
## team12                 -3.435e-02  1.418e-01  -0.242 0.808718
## targeted_productivity   3.705e+00  2.662e-01  13.921  < 2e-16 ***
## smv                    -3.903e-02  5.888e-03  -6.629 5.75e-11 ***
## wip                     1.178e-05  1.692e-05   0.696 0.486762
## over_time              -1.322e-05  1.177e-05  -1.123 0.261726
## incentive               2.609e-04  1.516e-04   1.721 0.085535 .
## idle_time               1.126e-03  2.213e-03   0.509 0.611107
## idle_men               -3.900e-02  9.550e-03  -4.084 4.81e-05 ***
## no_of_style_change     -1.785e-01  6.984e-02  -2.556 0.010753 *
## no_of_workers           2.257e-02  5.387e-03   4.190 3.06e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.763 on 926 degrees of freedom
## Multiple R-squared:  0.3208, Adjusted R-squared:  0.2988
## F-statistic: 14.58 on 30 and 926 DF,  p-value: < 2.2e-16
```

- RMSE value on validating lm model on test data set is as follows:-

```
lm_preds <- predict(fit_lm, newdata = test_set)
rmse_lm <- RMSE(lm_preds, test_set$rev_act_prod)
rmse_results<-data_frame(method="lm model",RMSE=rmse_lm)
```

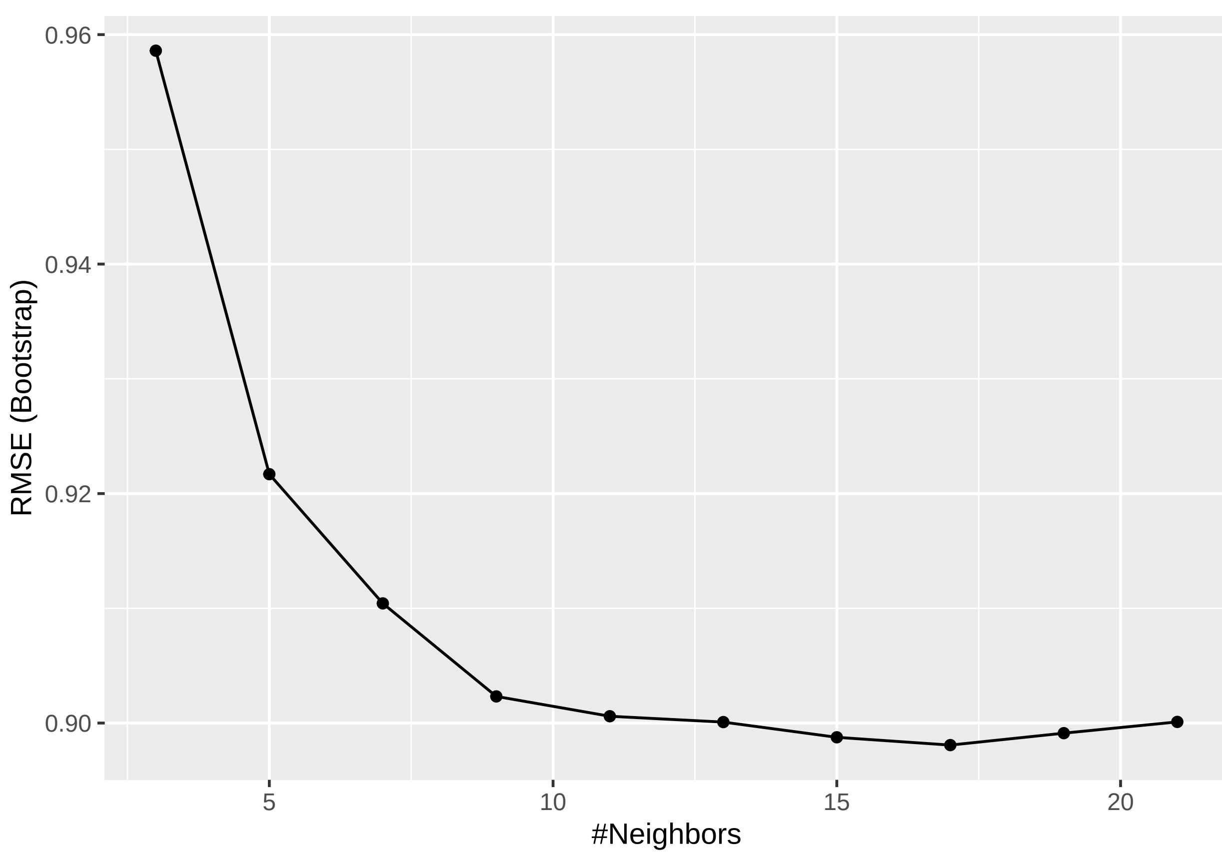
```
+ RMSE of Model:-
```

method	RMSE
lm model	0.6861627

- K-nearest neighbors model:- Used following code for building knn model

```
set.seed(7, sample.kind = "Rounding")
tuning <- data.frame(k = seq(3, 21, 2))
fit_knn <- train(rev_act_prod ~ .,data=train_set,
                 method = "knn",
                 tuneGrid = tuning)
```

```
+ plot of knn model gives best value of tuning parameter on which RMSE of training model is minimum.
```




```
+ Summary of knn model built
```

```
##     k      RMSE   Rsquared       MAE     RMSESD RsquaredSD      MAESD
## 1   3 0.9585994 0.09988784 0.6611679 0.03785403 0.03376719 0.03386466
## 2   5 0.9216919 0.10387981 0.6641705 0.03646603 0.03240918 0.02984845
## 3   7 0.9104291 0.10071463 0.6692934 0.03839413 0.03449856 0.02909561
## 4   9 0.9023217 0.09827545 0.6691845 0.03910566 0.03315079 0.02843797
## 5  11 0.9005933 0.09207551 0.6732006 0.03846723 0.03034220 0.02749475
## 6  13 0.9000842 0.08652578 0.6804792 0.03999827 0.03196580 0.02720898
## 7  15 0.8987585 0.08247769 0.6849436 0.03791958 0.03033622 0.02785140
## 8  17 0.8980788 0.07890931 0.6905333 0.03870749 0.02996676 0.02867593
## 9  19 0.8991176 0.07372396 0.6941995 0.03855279 0.02815282 0.02907897
## 10 21 0.9000990 0.06916352 0.6995258 0.03868100 0.02690402 0.02901873
```

`+ best value of tuning parameter is as follows:-`

```
fit_knn$bestTune
```

```
##    k
## 8 17
```

- Accuracy of the kNN model on the test set was calculated using following code:-

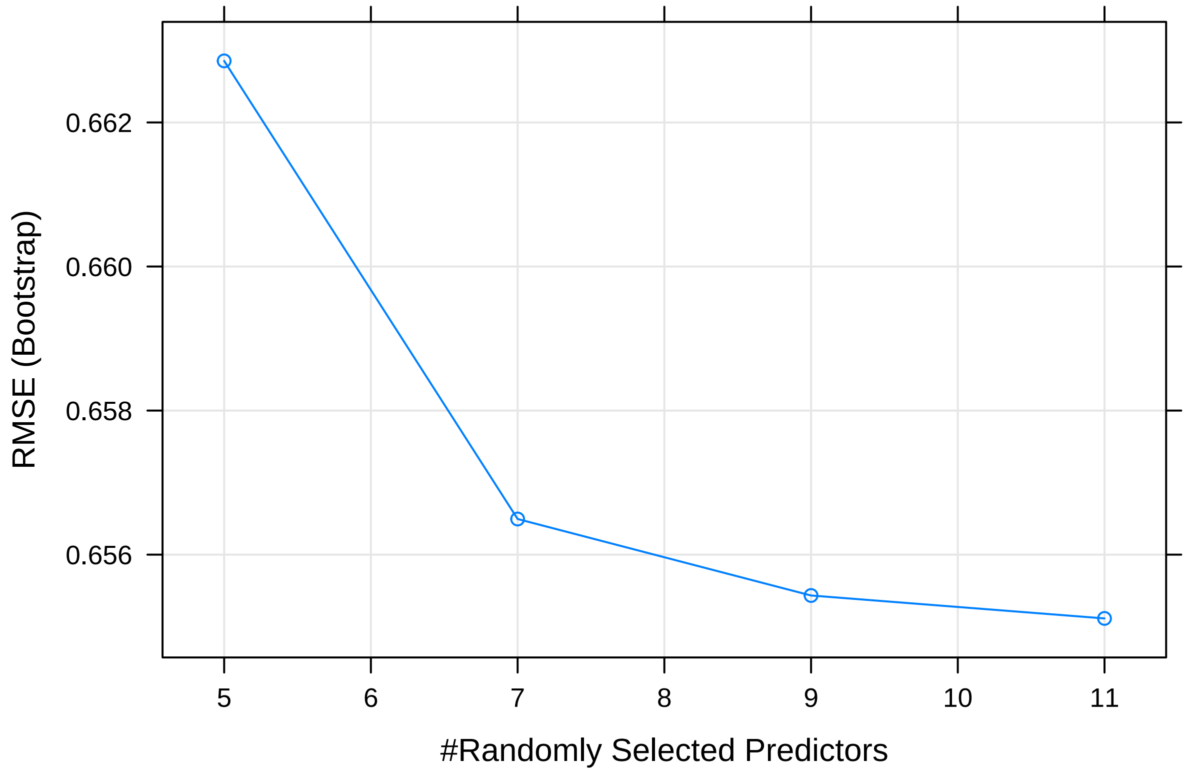
```
knn_preds <- predict(fit_knn, test_set)
rmse_knn <- RMSE(knn_preds, test_set$rev_act_prod)
rmse_results<-bind_rows(rmse_results,data_frame(method="knn_model",RMSE=rmse_knn))
rmse_results %>% knitr::kable()
```

method	RMSE
lm model	0.6861627
knn_model	0.7964137

- Random forest model:- Used below code for training the model-

```
set.seed(9, sample.kind = "Rounding")
tuning <- data.frame(mtry = c(5,7,9,11))
fit_rf <- train(rev_act_prod ~ .,data=train_set,
                method = "rf",
                tuneGrid = tuning,
               importance = TRUE)
```

- plot of random forest model tuning parameter where RMSE value is minimum for trained model

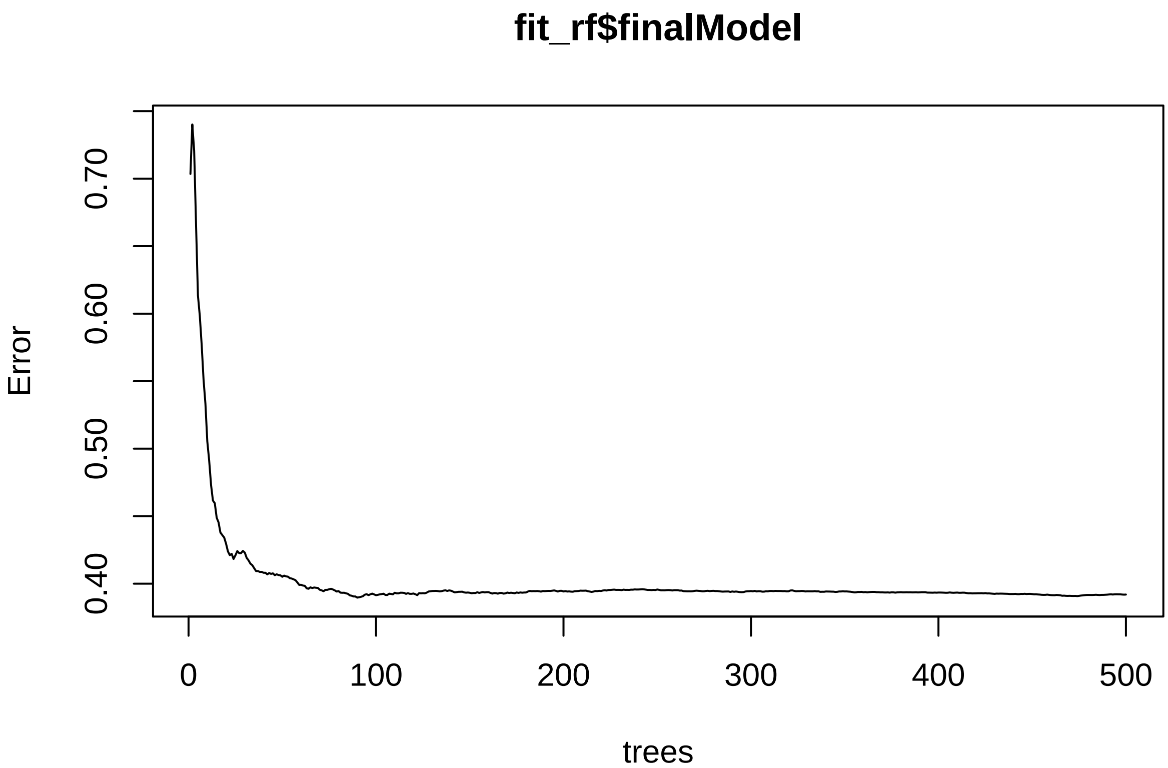


- Best value of tuning parameter is as follows:-

```
##   mtry
## 4   11
```

- Plot of final model shows that model trained has converged for given set of parameters as per following chart

```
plot(fit_rf$finalModel)
```



- The accuracy of the random forest model on the test set is estimated using RMSE formula as per following code:-

```
rf_preds <- predict(fit_rf, test_set)
rmse_rf <- RMSE(rf_preds, test_set$rev_act_prod)
rmse_results<-bind_rows(rmse_results,data_frame(method="rf_model",RMSE=rmse_rf))
rmse_results %>% knitr::kable()
```

method	RMSE
lm model	0.6861627
knn_model	0.7964137
rf_model	0.5954931

```
* Most important variable in the random forest model is obtained using following code. We can see incen
```

```
varImp(fit_rf)
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 30)
##
##                        Overall
```

```
## incentive              100.00
## targeted_productivity   99.47
## over_time               66.16
## smv                     59.83
## no_of_workers           58.01
## wip                     40.37
## quarterQuarter4         38.81
## quarterQuarter5         24.55
## team8                   24.25
## departmentsweing        23.38
## team11                  23.35
## quarterQuarter3         22.86
## idle_men                22.24
## no_of_style_change      21.50
## team12                  21.49
## team3                   20.49
## team9                   20.28
## idle_time               19.59
## team7                   15.88
## quarterQuarter2         15.87
```

- Summary of all the models:

```
## # A tibble: 3 x 2
##   method     RMSE
##   <chr>     <dbl>
## 1 lm model  0.686
## 2 knn_model 0.796
## 3 rf_model  0.595
```

Results

```
## # A tibble: 3 x 2
##   method     RMSE
##   <chr>     <dbl>
## 1 lm model  0.686
## 2 knn_model 0.796
## 3 rf_model  0.595
```

Results of machine learning algorithm shows that best model for predicting productivity of different function in garment sector is random forest model because it has lowest RMSE value.Further, productivity can be managed by incentive and targeted productivity.

Conclusion

Productivity is important parameter in labor intensive industry. Using this machine learning algorithm, management can take action proactively to avoid any situation in which business has to suffer because of order or revenue loss due to delay in order fulfillment because of low productivity environment.

```
+ Predicted productivity = 4 ==> Very High productivity ==> Low Priority
+ Predicted productivity = 3 ==> High Productivity      ==> OK
+ Predicted productivity = 2 ==> Poor Productivity      ==> Review
+ Predicted productivity = 1 ==> Very poor Productivity ==> High priority
```

Above table shows a scenario based on which management can decide what action to be taken so that productivity doesnt go down.