**Advanced Decision-Making Model for Predicting Garment Employee’s Productivity**



Table of Contents

[**1. ABSTRACT:** 3](#_Toc103374097)

[**2. ABBREVIATIONS & KEYWORDS:** 3](#_Toc103374098)

[**3. INTRODUCTION:** 3](#_Toc103374099)

[**4. METHODOLOGY:** 4](#_Toc103374100)

[**5. MAIN ELEMENTS:** 5](#_Toc103374101)

[5.1 Data Sources: 5](#_Toc103374102)

[5.2 Exploratory Data Analysis: 7](#_Toc103374103)

[5.2.1) Missing Values Handling: 7](#_Toc103374104)

[5.2.2) Date Object: 9](#_Toc103374105)

[5.2.3) Splitting based on Variable types: 10](#_Toc103374106)

[5.2.5) Month Field : 11](#_Toc103374107)

[5.2.6) Univariate Analysis: 11](#_Toc103374108)

[5.2.7 Bivariate Analysis: 12](#_Toc103374109)

[5.2.8) Correlation: 16](#_Toc103374110)

[5.2.9) Normality of Data: 17](#_Toc103374111)

[5.2.10) Scaling: 17](#_Toc103374112)

[5.2.11) Handling Outlier: 20](#_Toc103374113)

[5.3 Factor Analysis : 21](#_Toc103374114)

[5.4) Algorithms used in the Model: 22](#_Toc103374115)

[5.4.1) Regression: 22](#_Toc103374116)

[5.4.2) Classification: 24](#_Toc103374117)

[5.5) Evaluation Techniques: 25](#_Toc103374118)

[**6. RESULTS AND DISCUSSION:** 29](#_Toc103374119)

[**7.** **CONCLUSION:** 30](#_Toc103374120)

[**8.** **REFERENCE:** 31](#_Toc103374121)

[**9.** **R APPENDIX:** 31](#_Toc103374122)

# **1. ABSTRACT:**

The Garment industry is one of the significant industries in the recent times as its contribution to the economy is high. But at the same time, this particular industry is very slow in achieving its target because of the large number of processes are performed by the human beings. So, this has created a situation that the demand is increasing at one end and at the same time the targets are achieved at a slow pace. To handle this situation, advanced decision model needs to be applied on the Garment industry to predict the factors that makes the productivity grow. This paper analyses the Garment data to get meaningful insights from them. Also 6 machine learning models along with 4 evaluation metrics has been applied as well. On applying , it is evident that the KNN classifier and Multiple Linear regression are the best models in this decision making process.

# **2. ABBREVIATIONS & KEYWORDS:**

MAE – Mean Absolute Error , SVM – Support Vector Machine, EDA – Exploratory Data Analysis, KNN – K-Nearest Neighbours, RMSE – Root Mean Squared Error.

# **3. INTRODUCTION:**

Garment industry Plays an important role as garments are one of the essential things like food and shelter. This industry has contributed to the economy in a good way because of more employee has been recruited to the industry. The Garment production process is stepwise process like sewing, finishing and that needs to be carried out in the proper order and also careful consideration has to be made to get the end product. As the processes are complicated, the whole industry is purely dependent on the employees. To achieve the full production and good sales, the employee productivity needs to be tracked and also an analysis also needs to be carried out. As the Garment industry are making this process manually. The results are not accurate even after spending lot of time. As the technology emerges in recent days, some of the garment industry has realised the importance of applying the machine learning and the data science techniques to track and analyse the employee productivity. This will make the industry to achieve more target.

In recent times, the advanced decision making has been useful in almost all the industries. Their applications areas are numerous. Some of the examples are the Banking, Education, Health and Technology. Even though the machine learning and advanced decision making has been applied over major areas. The Garment industry has started adapting it quite late. On applying these techniques on the Garment industry, the underlying pattern on the data, the factors that are improving the productivity and the factors that are decreasing the productivity can be easily identified. Also, the necessary steps can also be taken to improve the productivity of the industry.

The structure of the paper can be demonstrated as : Section 4 explains the methodology in which the paper has been written, Section 5 gives detailed information like dataset source, data description, data cleaning, Exploratory data analysis, Univariate and Bivariate analysis, correlation and checking normality as well. Also, this explains about the 6 models applied and their evaluation metrics. Section 6 shows the result of the models and the discussions. Section 8 gives the conclusion of the paper.

# **4. METHODOLOGY:**

In this paper, proper workflow has been maintained as part of this model. The workflow begins with the Exploratory data analysis part followed by the training and test split of the data, factor analysis and finally the Evaluation models. The R language is used to write the code to develop this model. The below workflow gives a clear picture of the workflow of the paper.

**INTRODUCTION**

**METHODOLOGY**

**MAIN ELEMENTS**

**DISCUSSION**

**CONCLUSION**

**REFERENCES**

**EXPLORATORY DATA ANALYSISTIAL**

1. Data Cleaning
2. Factor Analysis
3. Outlier Detection
4. Missing Value Handling
5. Categorical and Continuous Variables
6. Visualisation of Data
7. Simple Linear Regression
8. Multiple Linear Regression
9. Logistic Regression
10. SVM
11. RF
12. KNN

**EVALUATION MODELS**

1. Precision
2. Recall
3. Confusion Matrix
4. RMSE
5. R-Squared
6. MAE

**ML MODELS**

# **5. MAIN ELEMENTS:**

The main elements of this paper for advanced decision making have been split into the below sections and subsections to explain the various techniques applied to the Garment data to Predict the productivity and also to categorize the data based on the Department factors. The major sections span around the process of cleaning the data and performing the exploratory data analysis on them.

Let us have a look at the detailed explanation of the above sections and their subsections.

## Data Sources:

This section aims to explain the source of the dataset used for the prediction of productivity. This dataset is extracted from the UCI Machine learning repository and has 1197 observations and 15 variables. The variables explain the details of the Garment employee’s work recorded in one of the industries. The date variable defines the date on which the data has been recorded and the day tells the day of the work and the quarter explains the Quarter of the year on winch the production happens. They are two departments; one is sewing and the other one is finishing. Also, the team number says the team which completes the production. Also, the number of workers explains the from each team who has completed the Garment production on that particular day. A number of style changes explains about the change of styles if applicable. The productivity field is the main focus area in this prediction. In that, the actual productivity shows the recorded and the targeted is the expected productivity of that garment. The idle time and men play a key role as they explain the idle time and the men work interrupted. Work in progress shows the uncompleted work. **Actual productivity is the Dependent variable for regression** and in the classification **department field is used to do categorization**.

Table 5.1 Attributes Explanation Source : uci

|  |  |  |
| --- | --- | --- |
| **Field No** | **Variables** | **Abbreviation** |
| 1 | date | Date in MM-DD-YYYY |
| 2 | day | Day of the Week |
| 3 | quarter | A portion of the month. A month was divided into four quarters |
| 4 | department | Associated department with the instance |
| 5 | team\_no | Associated team number with the instance |
| 6 | no\_of\_workers | Number of workers in each team |
| 7 | no\_of\_style\_change | Number of changes in the style of a particular product |
| 8 | targeted\_productivity | Targeted productivity set by the Authority for each team for each day. |
| 9 | smv | Standard Minute Value, it is the allocated time for a task |
| 10 | wip : Work in progress | Includes the number of unfinished items for products |
| 11 | over\_time | Represents the amount of overtime by each team in minutes |
| 12 | incentive | Represents the amount of financial incentive (in BDT) that enablesor motivates a particular course of action. |
| 13 | idle\_time | The amount of time when the production was interrupted due to  several reasons |
| 14 | idle\_men | The number of workers who were idle due to production interruption |
| 15 | actual\_productivity | The actual % of productivity that was delivered by the workers. It ranges from 0-1. |

The above table also explains the variable explanation in a precise manner. Initially the data is read in the data frame Garment and in the later sections, many data frames will be created based on the requirements further.

## Exploratory Data Analysis:

The Structure, head and summary of the data needs to be explored first before we apply the model because it will give the necessary information about the data. Also, the real time data that we extract will be a messy data with the uncleaned attributes and rows. So, first the data cleaning should be performed by using the EDA techniques. We can perform the below EDA techniques to make the data more interesting to apply to the models.

The basic commands like structure, summary and head(df) will display the structure of the data frame, a brief explanation of the Datatype and the first 6 rows of the dataset respectively. So, these commands were executed to get a overall view of the dataset. Next, we can look at the more interesting commands.

### 5.2.1) Missing Values Handling:

As part of the EDA process, we need to ensure that all the variables have data. Because feeding the missing value will be treated as Nan in the system and it will throw the error. So, the below piece of code to executed to get the graph plotted in the figure 5.2.1.1

|  |
| --- |
| #Missig value imputataion - Only WIP  apply(Garment, MARGIN = 2, FUN = function(x) sum(is.na(x)))  library(Amelia)  missmap(Garment, col = c("red", "blue"), legend = FALSE) |

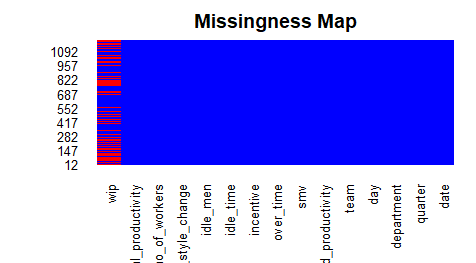
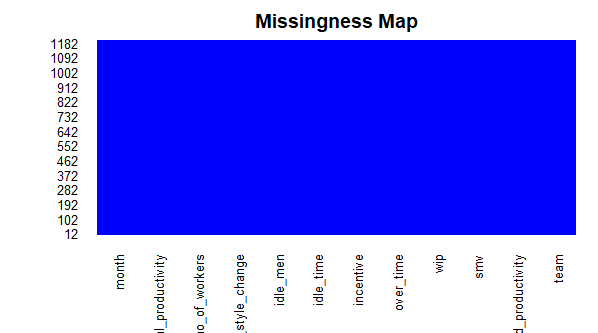


Fig 5.2.1.1 Missing Map

We could see that the column WIP has more missung value which needs to be treated . so , the below piece of code is applied to impute values in that.

|  |
| --- |
| library(dplyr)  library(zoo) #advanced date package  library(imputeTS)  G\_con$wip[is.na(G\_con$wip)] <- min(G\_con$wip, na.rm = T)  #After removal  apply(G, MARGIN = 2, FUN = function(x) sum(is.na(x))) |

The tidyverse package has been used to replace missing values with the minimum value. The mutate() function is used to specify the column the action should be applied. Then the apply function is used to replace the NA's with the smallest value of the wip column.After imputing , we could **see** that the missmap is generated as in below figure 5.2.1.2.

 Figure 5.2.1.2 Missing map after handling the missed variable

### 5.2.2) Date Object:

The output which will be dispalyed on running the str(Garment) data farme has been shown in figure 5.2.2.1. We could see that the Crime data frame has a data attribute which is incorreclty named as chr. So, we need to change that as data attribute.

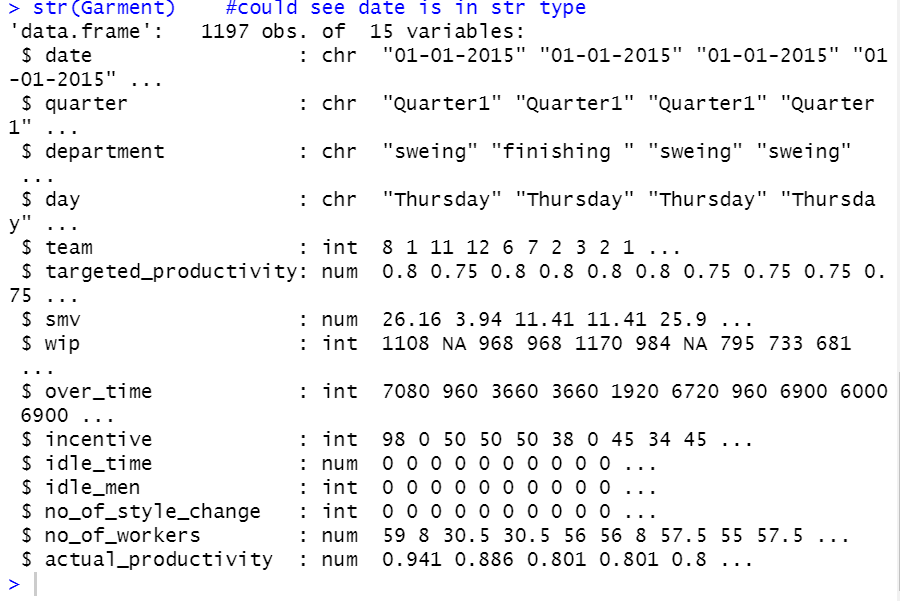


Figure 5.2.2.1 Structure of Data frame

The below piece of code has been used to achieve the conversion.

|  |
| --- |
| a <- as.Date(Garment$date,format="%d/%m/%Y") # Produces NA when format is not "%d-%m-%Y"  b <- as.Date(Garment$date,format="%d-%m-%Y") # Produces NA when format is not "%d/%m/%Y"  a[is.na(a)] <- b[!is.na(b)] # Combine both while keeping their ranks  Garment$date <- a # Put it back in your dataframe |

The function as.Date is used to achieve this, In the input dataframe , the date format were of two formats so the function is applied twice by **changing the format parameter**. Now, date attribute is successfully converted as date object.

### 5.2.3) Splitting based on Variable types:

As per figure 5.2.2.1, it is clear that we have both categorical and continuous variables. So, it will be very useful to separate them in different data frames to treat them accordingly. So, the Garment dataframe is splitted as G\_cat and G\_con which contains the categorical and continuous separately.The code used is attached below:

|  |
| --- |
| #Separate Categorical and Continous variables  G\_con = Garment[ ,!sapply(Garment, is.character)]  G\_cat = Garment[ ,sapply(Garment, is.character)] |

The function is.Character is used to achieve this. This function will return the char variable. So in first line of command , the numeric is taken by using the not(!) function and in the second line of the code , the character variables are taken into the dataframe.

5.2.4) Department Field Cleaning:

The Department field has three values but there is a extra space after the variable finishing which makes it as a separate class. So, the whitespace has been trimmed and the spelling mistake of the sewing variable has been corrected as well by using the below code:

|  |
| --- |
| G\_cat$department <- trimws(G\_cat$department)  G\_cat$department <- na.omit(G\_cat$department)  library("stringr")  G\_cat$department = str\_replace\_all(G\_cat$department,'sweing','sewing') |

The stringr Package is used for using str\_replace\_all whereas str\_replace will only do the first occurance.

### 5.2.5) Month Field :

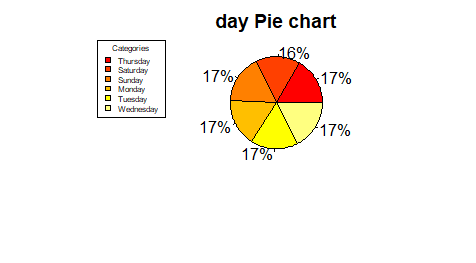
The months function is used to convert the date attribute to month attribute so that it fetches the month variable and displays them as a separate field in the same dataframe.

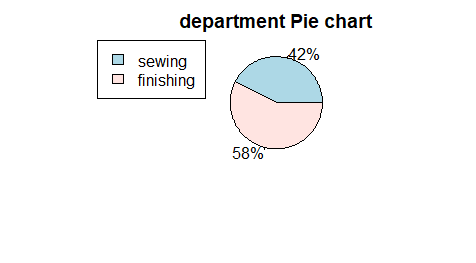
|  |
| --- |
| G\_con$month <- months(as.Date(G\_con$date)) |

### 5.2.6) Univariate Analysis:

Univariate analysis is that in which the only one attribute’s data is plotted and the inference is labelled. The univariate analysis is carried out for 3 categorical fields (Day , Quarter and the Department). Pie chart is a chart which displays the field as categories and the count of the values as proportion. The code used is shown below:

|  |
| --- |
| pie(table(G\_cat$quarter), labels = paste(round(prop.table(table(G\_cat$quarter))\*100), "%", sep = ""),  col = heat.colors(5), main = "Quarter Pie chart")  legend("topright", legend = c("Quarter1", "Quarter2", "Quarter3", "Quarter4", "Quarter5"),  fill = heat.colors(5), title = "Categories", cex = 0.5) |





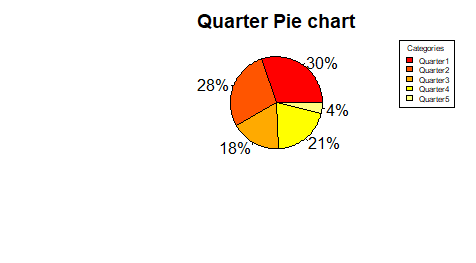


Fig 5.2.6.1 All Pie charts

**The inference is shown below:**

The quarter4 does not have data so there may be reason behind that. In that case, we can assume that **Quarter5 is a holiday quarter**. Finishing department has slightly large amount of data comparative to the sewing data but they are mostly equal in proportion. So, we need not add or reduce values in their category. **Friday is the weekly off** and we have data for all the other days. These inference from the data will be useful in the future prediction.

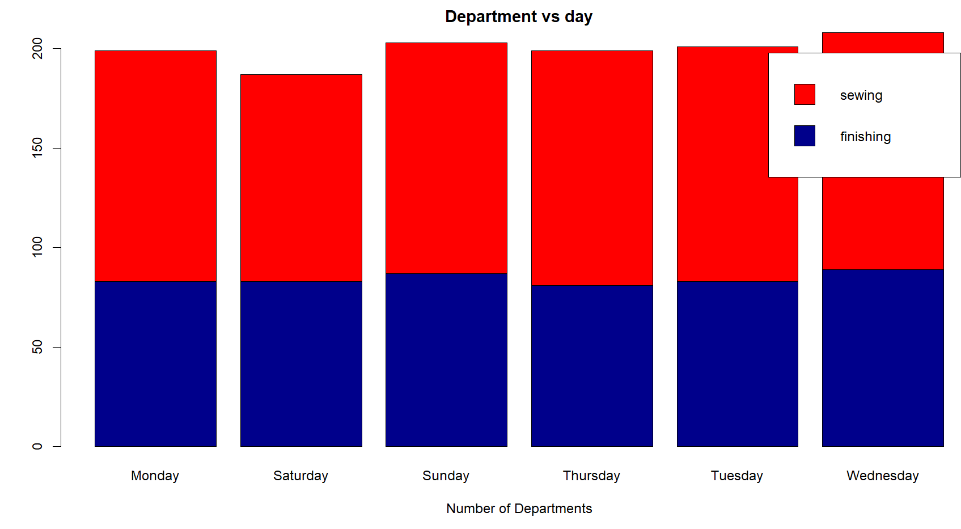
### 5.2.7 Bivariate Analysis:

Bivariate analysis is analysing two variables at a time. We could see that the Sunday has more finishing and the sewing. Saturday has the less finishing and sewing out of all the days in the week. The other days have the moderate same range of production with respect to the department.

#### 5.2.7.1 Bar charts:

Bar chart is a chart which displays each field as different bars. The code used is shown below:

|  |
| --- |
| counts <- table(G\_cat$department,G\_cat$day)  barplot(counts, main="Department vs day",  xlab="Number of Departments", col=c("darkblue","red"),  legend = rownames(counts)) |



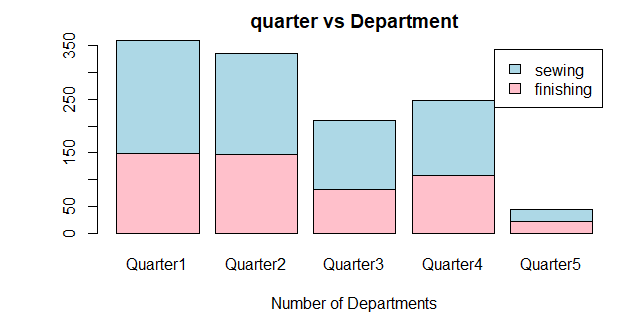


Fig 5.2.7.1 All Bar charts

As we assumed in the univariate analysis that the Quarter5 is a holiday off and it is confirmed here as well. Also , the Initial Quarter has more production and the third quarter has the least production.

#### 5.2.7.2 Histograms:

Histogram shows how frequent a value occurs in an attribute. Also, it displays them in form of range. All the continuous variables have been plotted in the histogram to look at their frequency of occurrence in the dataset.

**The inferences are shown below:**

Idle men, Idle time and number of style change attributes have only the value Zero. So, we need not use it because it is not going to impact the productivity in the model.

The targeted and the actual productivity has the same range of values. So, we can consider them as likely attributes.

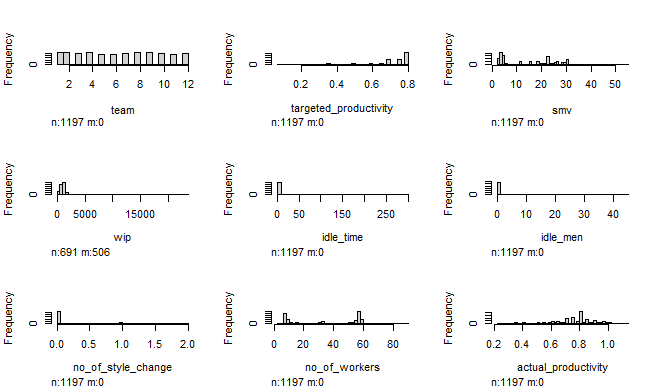


Fig 5.2.7.2 Histogram of Continuous variables

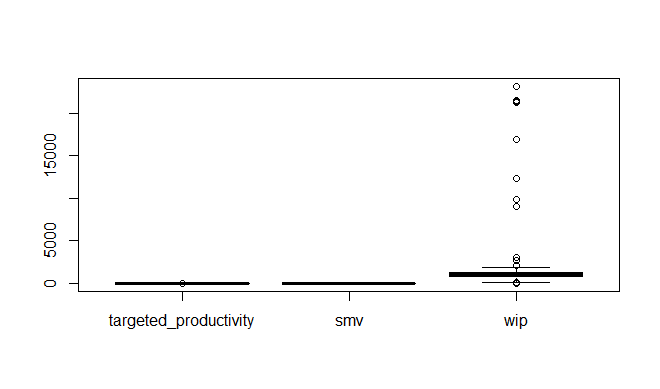
The code used is shown below:

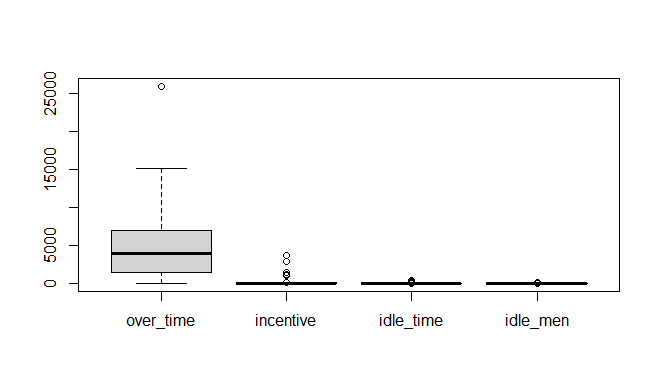
|  |
| --- |
| library(Hmisc)  hist.data.frame(G\_con[c(-5,-6)])  hist.data.frame(G\_con[c(2,3,4,5,6)]) |

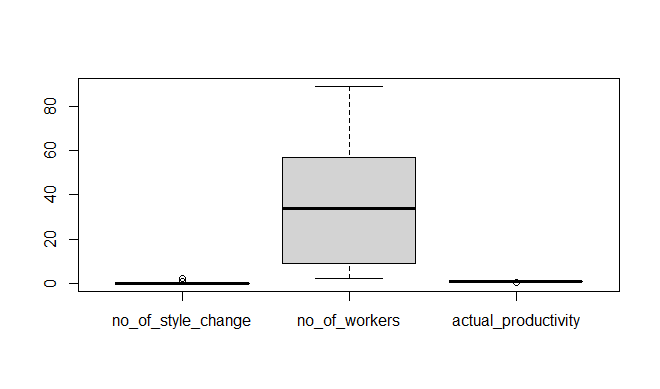
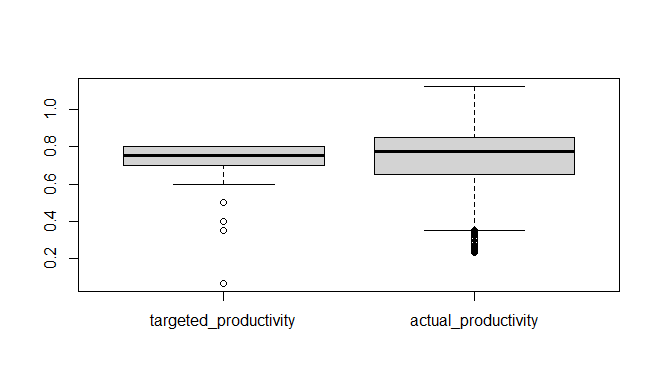
#### 5.2.7.3 Boxplot:

The boxplot shows the outlier values as rounded form and the inter quantile, and percentile values as well. This is very effective to know whether all the variables are scaled equally. Also, to know whether there are any outliers so that we can handle them accordingly.

Figure 5.2.7.3 All boxplots





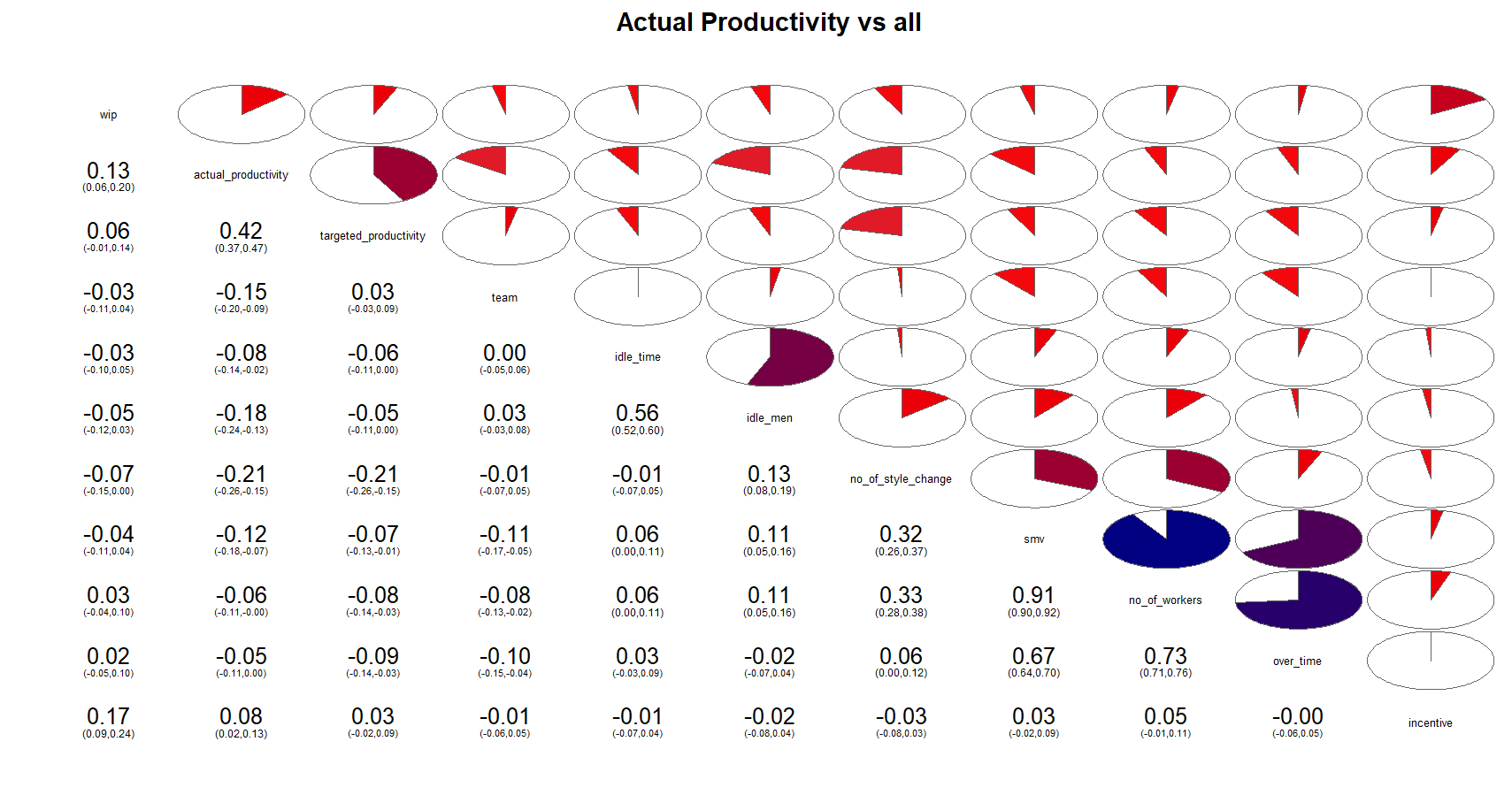


We could see that the target productivity value stopped at 0.8 but actual is more than 1. So they are not standard. The code used is as follows:

|  |
| --- |
| boxplot(G\_con[c(2,3,4)])  boxplot(G\_con[c(5,6,7,8)])  boxplot(G\_con[c(9,10,11)]) |

### 5.2.8) Correlation:

The correlation graph displays a value which shows the amount of corelation a variable holds with the other variable. In the result, a positive correlated value depicts that if one variable’s value gets higher, the another value gets higher as well. Also, the more the value close to 1, the more they are correlated. A negative correlated value depicts that if one variables’ value gets higher, the another value gets lower



From the above figure, we can see that there is very high correlation between smv and no\_of\_workers. Also, over\_time & smv as well as over\_time and no\_of\_workers are correlated. With respect to the dependent variable targeted productivity , the highly correlated independent variable is the actual productivity. Please refer the code below:

|  |
| --- |
| library(corrgram)  # corrgram works best with Pearson correlation  corrgram(G\_con, order=TRUE, cor.method = "pearson", lower.panel=panel.conf,  upper.panel=panel.pie, text.panel=panel.txt, main="Actual Productivity vs all",  col = colorRampPalette(c("#8073AC", "red", "navy"))) |

### 5.2.9) Normality of Data:

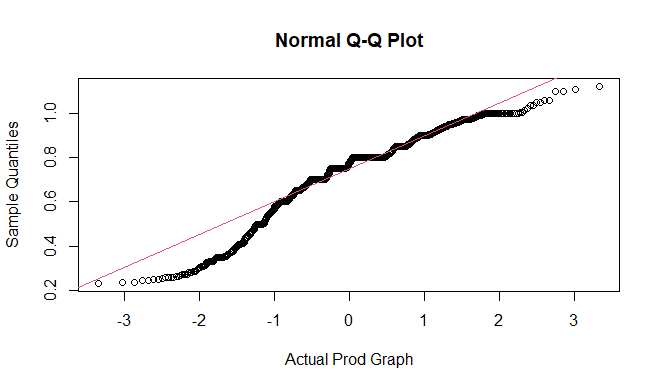
Before applying the linear regression, the dependent variable should be normally distributed. So, we need to plot them in q-q plot (If the data points are in the straight line, then the data is normally distributed otherwise it is not.

Figure : 5.2.9 QQ plot for independent variable.

We need to apply the scaling technique to resolve this and that will be covered in the next section of the paper. Please find the code for the QQ plot below:

|  |
| --- |
| qqnorm(Garment\_f$actual\_productivity, xlab = "Actual Prod Graph" )  qqline(Garment\_f$actual\_productivity, col = 2) ## red color |

### 5.2.10) Scaling:

Scaling is a technique of normalising all the values of the variables in the same range and there are several scaling techniques available in R and we need to figure out the appropriate technique which scales out the variable equally. Let us see the scaling techniques applied in our model. Please find the figure 5.2.10.a which shows the box plot before applying any scaling technique.

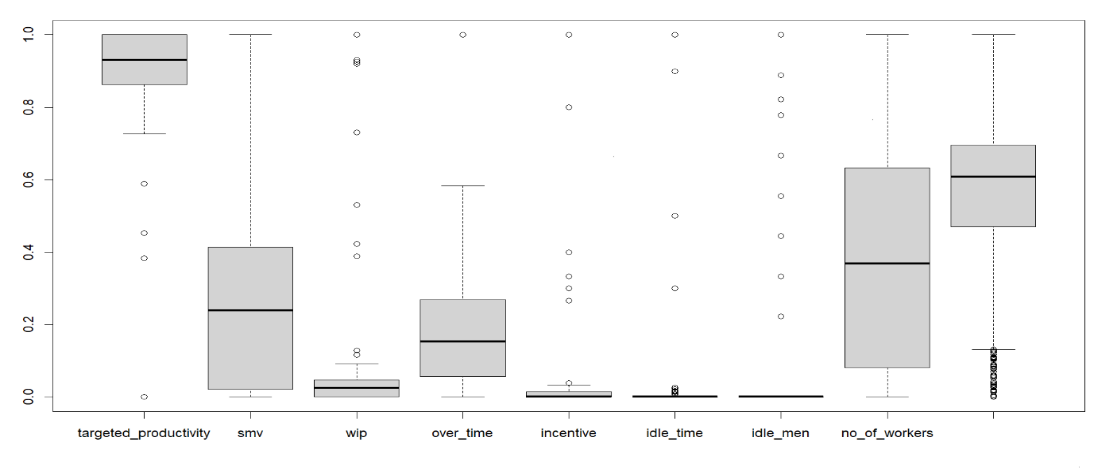


Fig 5.2.10.a Before Scaling

#### 5.2.10.1) Softmax :

Softmax is a scaling technique in which the values are scaled based on the lambda value, average and the standard deviation values. Please find the code and the figure below for softmax scaling.

|  |
| --- |
| Garment.sm <- apply(Garment.z1, MARGIN = 2, FUN = function(x) (SoftMax(x,lambda = 1, mean(x), sd(x))))  boxplot (Garment.sm, main = "Soft Max, lambda = 1") |

I tried with different lambda value and could see that they were not scaled properly.

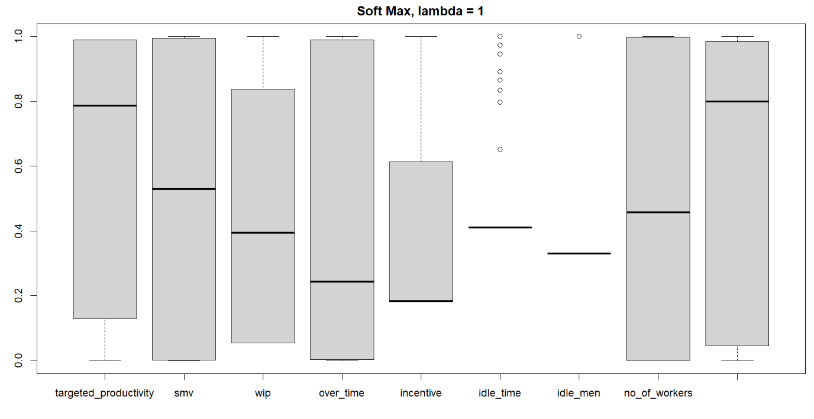


Fig 5.2.10.1 Lambda=1 Softmax

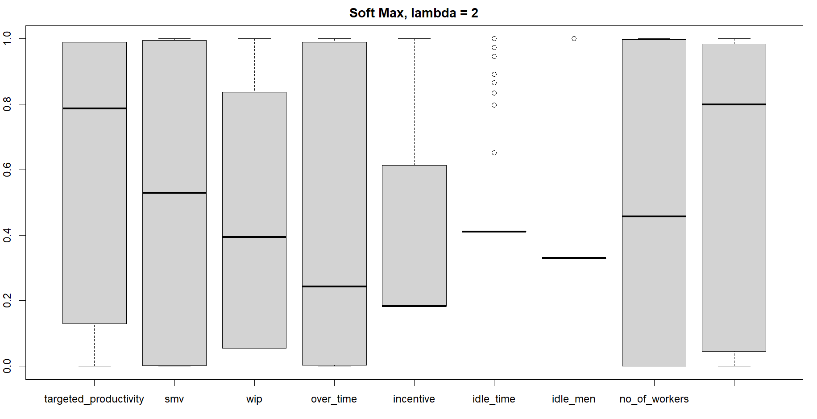


Fig 5.2.10.1 Lambda=2 Softmax

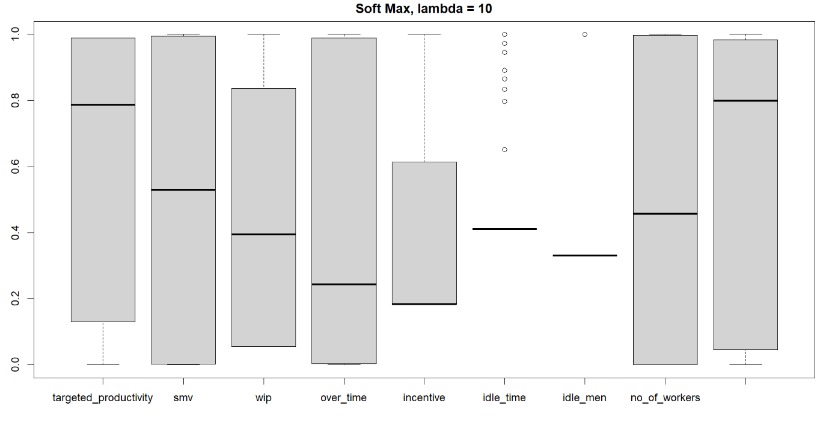


Fig 5.2.10.1 Lambda=10 Softmax

#### 5.2.10.2) min-max:

Min-max scaling technique applies the formula x - min(x))/diff(range(x) and this scales the value based on the minimum and the maximum values. Please find the code below:

|  |
| --- |
| Garment.mm <- apply(G\_connew[], MARGIN = 2, FUN = function(x) (x - min(x))/diff(range(x))) |

#### 5.2.10.3) Z-score:

Z-score tells the number of deviations from a point called mean. If it is Zero, then the mean value and the data points are equal.

|  |
| --- |
| Garment.z1 <- apply(Garment.mm, MARGIN = 2, FUN = function(x) (x - mean(x))/sd(x)) |

On applying with one and two standard deviation and the values are scaled evenly.

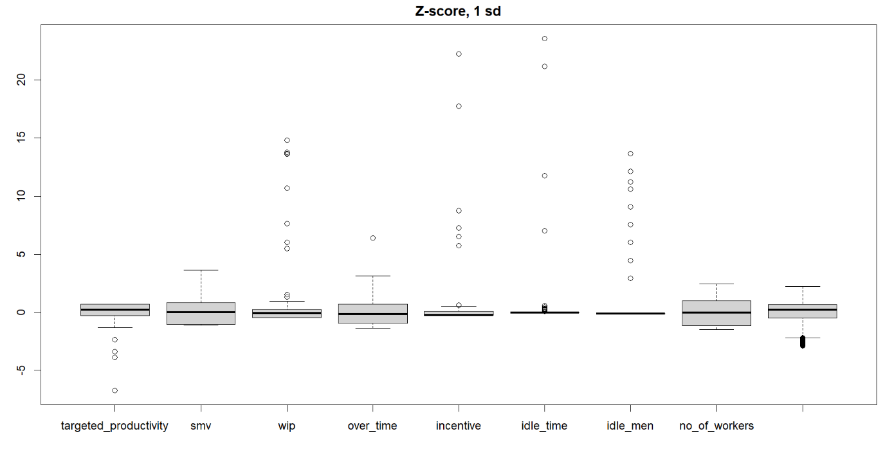


Fig 5.2.10.3 Z-score SD=1

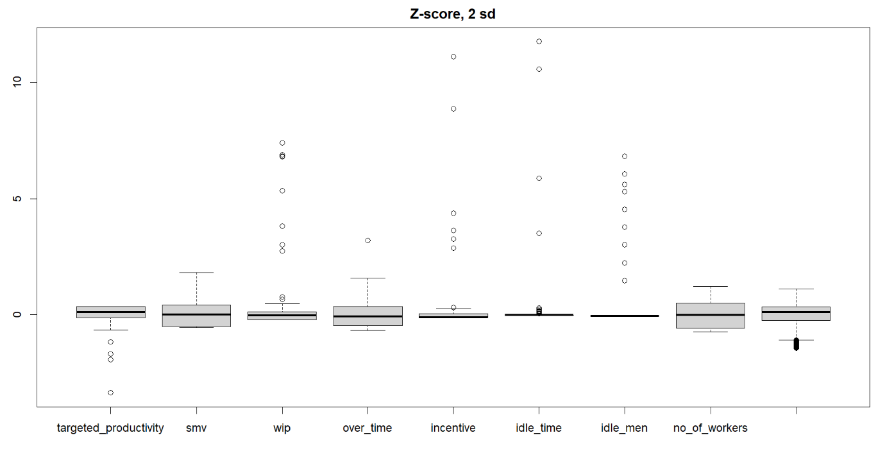


Fig 5.2.10.3 Z-score SD=2

So, after scaling the values, the QQ plot is shown as below and it is clear that the values are in straight line. So, they got scaled equally.

**After Scaling:**

The below QQ plot and the scatter plot has been plotted after scaling the values.

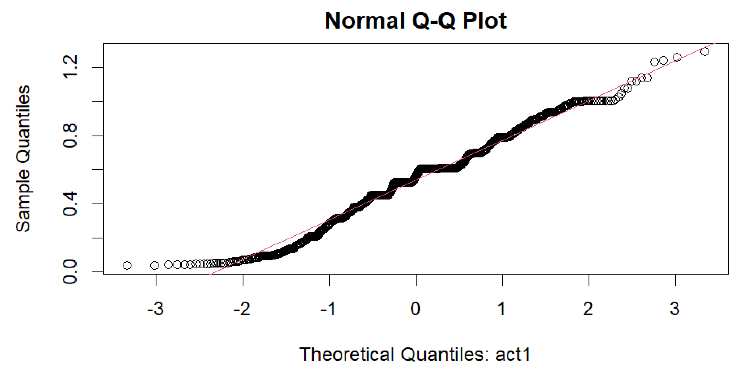


Fig 5.2.10.1 QQ-After Scaling

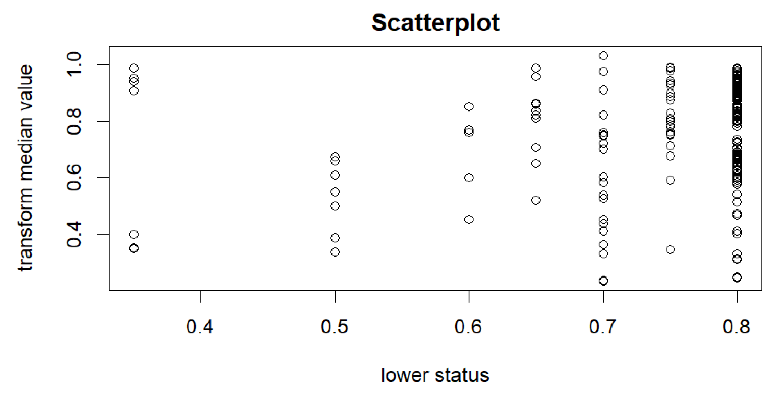


Fig 5.2.10.2 Scatter Plot - After Scaling

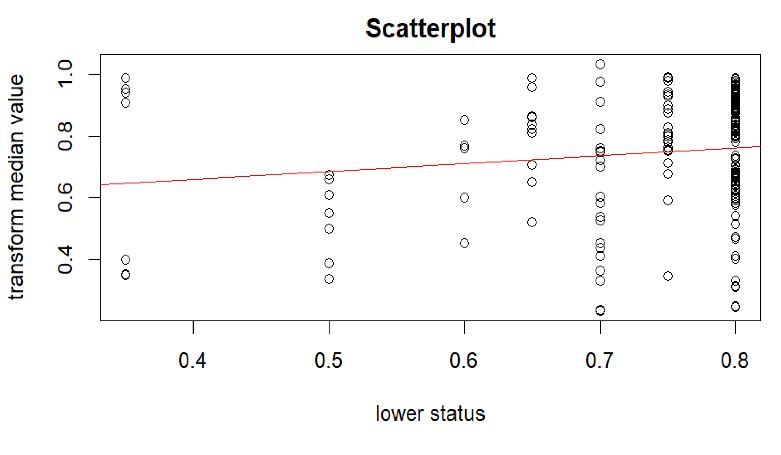


Fig 5.2.10.3 Scatter Plot – with line

### 5.2.11) Handling Outlier:

Outliers are values which are extreme either in terms of high value or low value. In the figure 5.2.10.3, there were outlier values available and they can be removed by applying the below piece of code.

|  |
| --- |
| replace\_outlier <- function(x){  for (i in which(sapply(x, is.numeric))) {  quantiles <- quantile( x[,i], c(.05, .95 ), na.rm =TRUE)  x[,i] = ifelse(x[,i] < quantiles[1] , quantiles[1], x[,i])  x[,i] = ifelse(x[,i] > quantiles[2] , quantiles[2], x[,i])}  x} |

### 5.2.12) One-hot Encoding:

One-hot encoding is the technique in which the character variables are changed into continuous values which is easy to apply in the regression models. i.e., If the field department has values as sewing and finishing, after applying the one hot encoding code, it will be converted as 1 for sewing and 0 for finishing. We need to drop one column as the other one clearly serves the purpose alone. Otherwise, there will be a dummy trap in the model.

|  |
| --- |
| library(caret)  dummy <- dummyVars(" ~ .", data=G\_cat)  G\_catenc <- data.frame(predict(dummy, newdata = G\_cat)) |

## Factor Analysis :

Factor analysis is a process used in advanced decision models to predict the number of optimised factors.The nfactors library is used to find the number of factors to take into account. Also, the scree plot is a plot which clearly shows the number of components by taking eigen values into account. The point where the scree plot becomes constant can be decided to find the number of components. Please refer the figure 5.3.1 for scree plot.

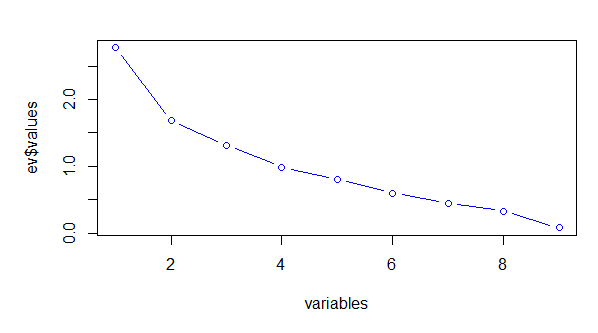


Figure 5.3.1 scree plot

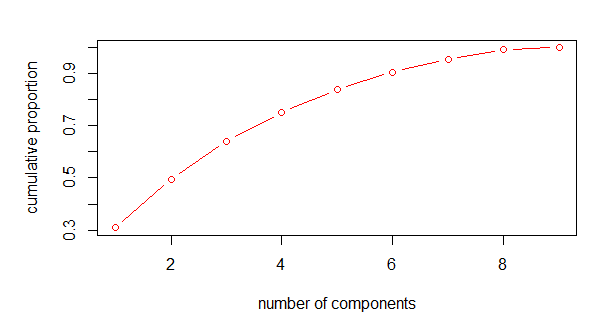


Fig 5.3.2 Cumulative proportion

The graph shows the variance percentage and the number of factors that can be considered and it is clear from the above graph that six factors can be considered. The model can be trained with 6 factors. So, the multiple regression is trained with 6 highly correlated factors. The correlated factors can be referred in the corrgram output.

## 5.4) Algorithms used in the Model:

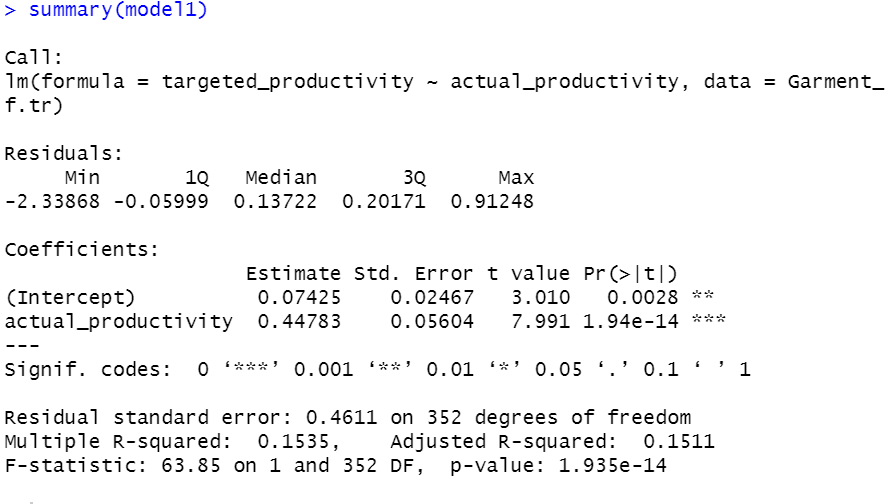
After Scaling the values, the below regression techniques are applied to predict the actual productivity and the best model is evaluated in section 5.5. Also, the classification algorithms are also applied to find the department of each production provided the other factors.

### 5.4.1) Regression:

Regression is a technique in which the dependent variable(actual productivity) is predicted based on the independent variables. There are several assumptions of this algorithm which needs to be verified before we apply them. The data needs to ne normal and also linearity, variance and independence of the data has to be checked. There are various types of regression available and now we are going to implement two techniques. They are discussed in the below subsections.

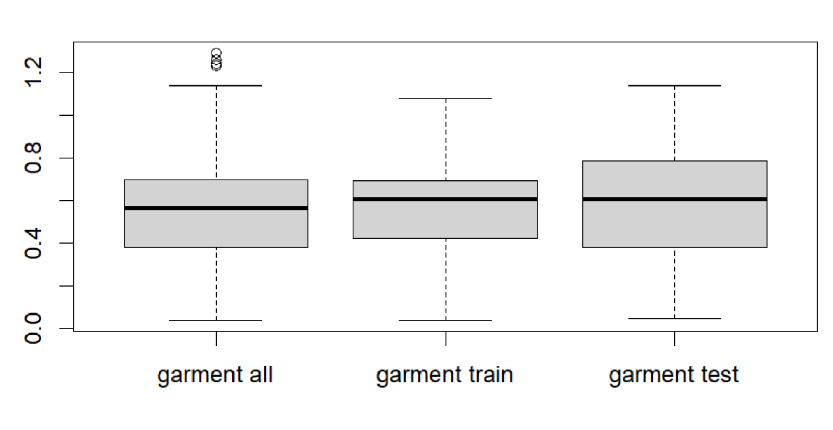
#### 5.4.1.1) Simple Linear Regression:

Simple Linear regression is a technique in which only one independent variable is applied to predict the dependent variable. In our model, the variable targeted productivity is used to predict the actual predictivity and the results are discussed in section6.



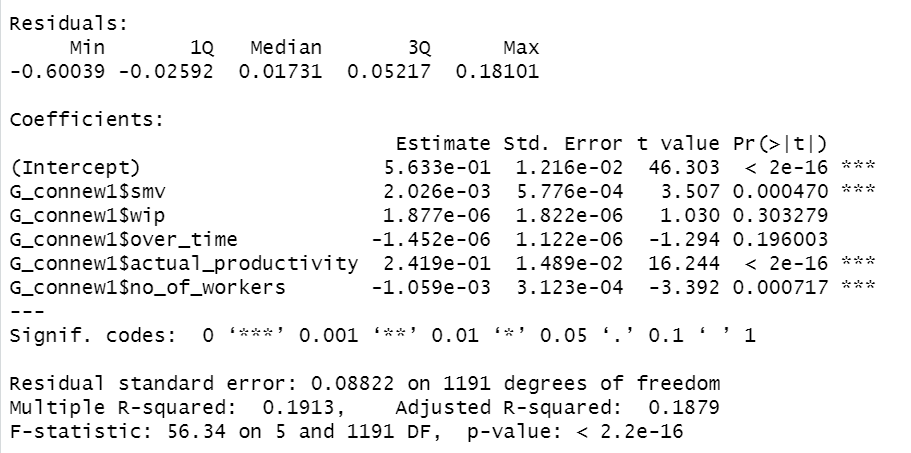
Train-Test-Split:

The input has been split into 70 and 30 percentage and splitted as train and the test and the boxplot has been shown below. The train data has been applied into the model and the test data has been predicted.



#### 5.4.1.2) MLR - Multiple Linear Regression:

Multiple Linear regression is a technique in which many independent variables are applied to predict the dependent variable. In our model, the variables svm, wip, overtime, targeted productivity and no of workers are used to predict the actual predictivity and the results are discussed in section6.



### 5.4.2) Classification:

In advanced decision making, classification process is a task in that a existing label series is identified for the new data point. This model tries to find conclusion on the training set data with known labelled records. In this paper, we have applied 4 different classification algorithm which is explained in below subsections.

#### 5.4.2.1) KNN:

K nearest neighbour is the algorithm which belong to the supervised category and it categorise a new data to the already existing point with respect to the feature similarity. The optimsed K-value which is selected after applying the cross table is 1. Cross table needs to be created before applying the model. Please find the code below:

|  |
| --- |
| Gar\_test\_pred <- knn(train = Gar\_train, test = Gar\_test,  cl = Gar\_train\_labels, k=1) |

#### 5.4.2.2) Random Forest:

Random forest (Imran, 2021), is great algorithm which is a bunch of decision trees. This works by creating more tress with the help of many factors. Please find the code below:

|  |
| --- |
| model.rf = train(department ~ .,  data = Gar\_trainrf,  method = "rf",  trControl = fitControl) |

#### 5.4.2.3) Logistic Regression:

Logistic regression deals with the binary outcome. In this model, as the department has only two values. We have applied this as well. This model uses a sigmoid function to predict the class of the variable. Even though the term regression is used, this is basically a classification algorithm and this comes under the supervised category. Please find the code below:

|  |
| --- |
| model.lr = train(department ~ .,  data = Gar\_trainlr,  method = "glm",  trControl = fitControl) |

#### 5.4.2.4) SVM:

SVM works well with both linear and nonlinear data points as it has the support vetors. SVM is basically a line drawn with the slope area around it. As the data is linear , here the kernel type is passed as linear.

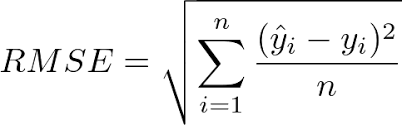
|  |
| --- |
| model.svm = svm(formula = department ~ .,  data = Gar\_trainlr[1:6],  type = 'C-classification',  kernel = 'linear') |

## Evaluation Techniques:

After applying the model on the training data, we have predicted the test data. Now, it is the time to check whether the predicted values and the already existing values are similar. If it is the same, then our model works fine. Otherwise, it is not accurate. Confusion matrix has been used for the classification algorithms to represent the classification results. We have also used other different evaluation metrics namely RMSE, MAE and R-Squared. The applicable evaluation metrics are described as follows:

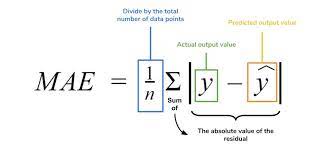
#### 5.5.1) RMSE :

RMSE stands for Root Mean Squared Error and is used in the regression algorithm. Residuals are the errors. This calculates the deviation of these residuals from the linear regression line. This also says how well the line is fitted to the model. The lesser the value , the more good the model is.



#### 5.5.2) MAE :

MAE is the mean absolute error. This is also applied on linear regression and the actual and the predicted values are subtracted and finally divided by the number of the data points. This value should be a low value and it means that the model is a robust one.



#### 5.5.3) R-Squared :

R-Squared is the total of squares of the error divided by the initial sum of errors and also subtracted the whole value from 1. This works well in case of handling the negative values as well. This value should be less and it means the model is optimised.

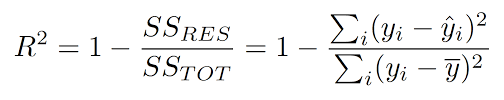
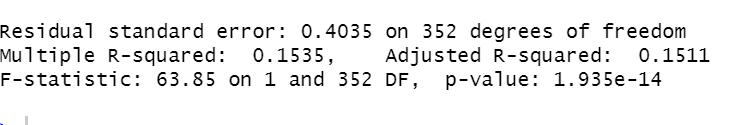


Fig 5.5.1.1 Evaluation using R-Squared Method - Formula



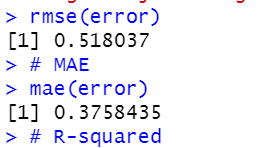
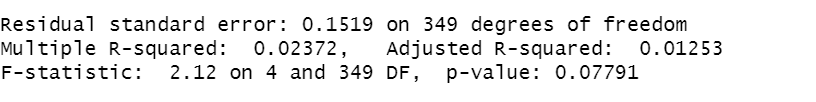
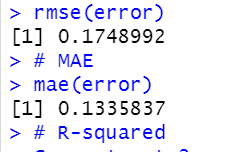


Fig 5.5.1.2 Evaluation- Multiple Linear





#### 5.5.4) Confusion Matrix:

This matrix is used as an evaluation metric in the classification algorithm. This displays the matrix between the true class and the predicted class. In the below figure, the count in the green area should be more and in the red area, the value should be less. TP denotes True Positive, TN means True Negative, FP means False Positive and FN means False Negative.

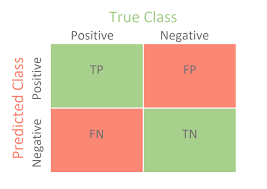
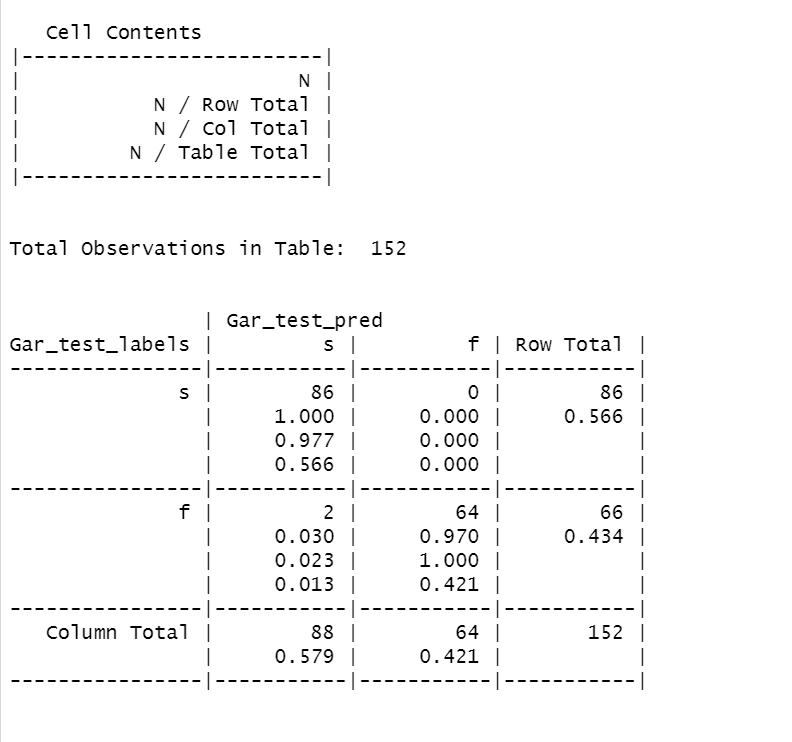


Fig 5.5.4. Confusion Matrix

The below figures show the screen-prints of the evaluation metrics applied on the different models.

Fig 5.5.2.1 KNN – Classification



Our confusion matrix, pays the way for the below metrics which is displayed in the below figure.

|  |
| --- |
| Accuracy (all **correct** / all) = TP + TN / TP + TN + FP + FN  Precision (**true** positives / **predicted** positives) = TP / TP + FP  Sensitivity aka Recall (**true** positives / all **actual** positives) = TP / TP + FN  Specificity (**true** negatives / all **actual** negatives) =TN / TN + FP |

Fig 5.5.2.2 RF - Classification

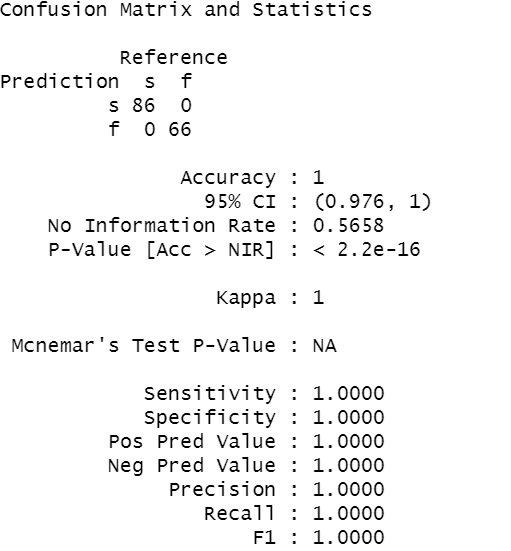


Fig 5.5.5.3 Logistic Regression – Classification

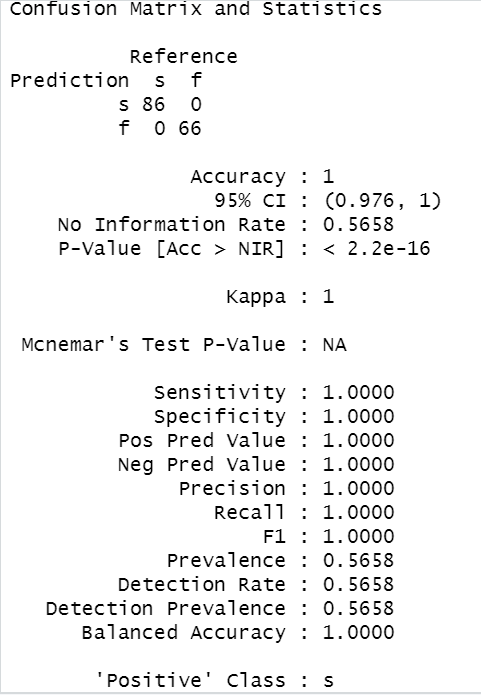
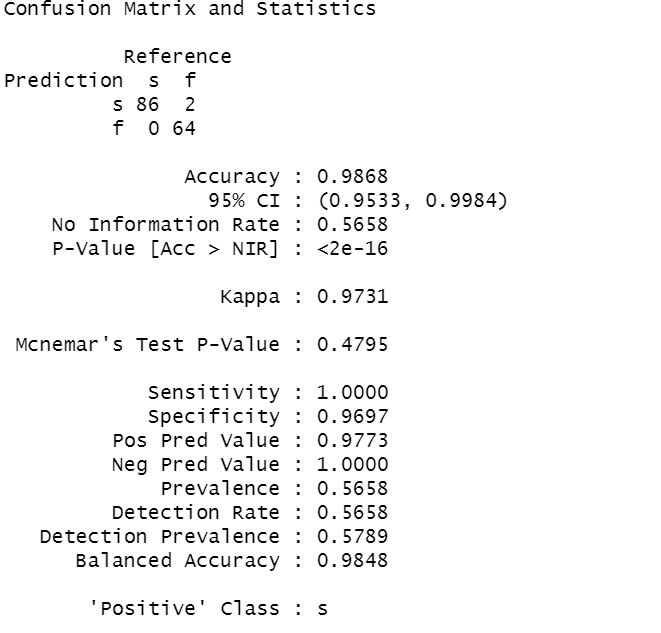


Fig 5.5.5.4 SVM – Classification



# **6. RESULTS & DISCUSSION:**

In this part of the paper, the results of the various regression models are presented in the table form and the best model is decided with respect to evaluation metrics of both the regression and the classification algorithm as well.

**Regression Results:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **MAE** | **RMSE** | **R-Squared** | **Adjusted R-Squared** |
| **Simple** | 0.518 | 0.375 | 0.1535 | 0.1511 |
| **Multiple** | 0.174 | 0.133 | 0.023 | 0.0125 |

Table 6.1 Regression

As per the regression result, **the Multiple linear regression works well** with the less R-Squared value and the adjusted R-Squared value. This means that 99.97% of the prediction is right. So Multiple linear regression works well.

**Classification Results:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **True Positive** | **True Negative** | **False Positive** | **False Negative** | **Sensitivity** | **Specificity** | **Accuracy** | **Precision** | **Recall** |
| **KNN** | 86 | 64 | 0 | 2 | 0.9694 | 1 | 0.9868 | 1 | 0.9773 |
| **Random Forest** | 86 | 66 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| **Logistic** | 86 | 66 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| **SVM** | 86 | 64 | 2 | 0 | 1 | 0.9697 | 0.9868 | 0.9773 | 1 |

Table 6.2 Classification

As per the classification result, the models Random Forest and Logistic Regression is perfect. But there may be chances that it will be overfitting of the model. So, **the best model is KNN** because a model should be more specific than sensitive with respect to the garment industry.

# **CONCLUSION:**

In this paper, our target was to find the factors that makes the productivity grow in the emerging garment industry. To handle this problem, we have used 6 different advanced decision-making techniques, namely K-nearest neighbours, logistic regression, support vector machine, random forest, linear regression and multiple linear regression on a dataset that was gathered from UCI machine learning repository. The model performances were evaluated thoroughly to find out the best optimised model. The dataset has been divided into two proportions (70% for training and 30% for testing) . The model has been trained on the training data and the prediction has been done on the testing data. Then after comparing the predicted and the existing values the model performance has been evaluated. Our experiment showed that, for predicting the department class, k nearest neighbour with k value as 1 produces the highest accuracy (= 98.68%). Furthermore, the multiple linear regression with smv, incentive, targeted productivity, actual productivity produces the best result accuracy (= 99.97%).

In the near future, this model can be enhanced by applying various techniques including the neural networks and the bagging techniques as well to gain some other meaningful insights from the data.

# **REFERENCE:**

1. Imran, Abdullah Al & Rahim, Md Shamsur & Ahmed, Tanvir. (2021). Mining the Productivity Data of Garment Industry. International Journal of Business Intelligence and Data Mining. 1. 10.1504/IJBIDM.2021.10028084.
2. A. A. Imran, M. N. Amin, M. R. Islam Rifat and S. Mehreen, "Deep Neural Network Approach for Predicting the Productivity of Garment Employees," 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), 2019, pp. 1402-1407, doi: 10.1109/CoDIT.2019.8820486.
3. Obiedat, R. and Toubasi, S.A., 2022. A Combined Approach for Predicting Employees’ Productivity based on Ensemble Machine Learning Methods. Informatica, 46(5).
4. Goyzueta, C.A.R., De la Cruz, J.E.C. and Machaca, W.A.M., 2021, October. Advantages of Assembly Machine Learning Models for Predicting Employee Productivity in a Garment Manufacturing Company. In 2021 IEEE Engineering International Research Conference (EIRCON) (pp. 1-4). IEEE.
5. Balla, I., Rahayu, S. and Purnama, J.J., 2021. Garment Employee Productivity Prediction Using Random Forest. Techno Nusa Mandiri: Journal of Computing and Information Technology, 18(1), pp.49-54.

# **R APPENDIX:**

R Code :

