Project Report
On
Employee Absenteeism

Vismay Dhobe

12th August 2018

Contents

1. Introduction

1.1 Problem Statement:

XYZ is a courier company. As we appreciate that human capital plays an important role in collection, transportation and delivery. The company is passing through genuine issue of Absenteeism. The company has shared it dataset and requested to have an answer on the following areas:

1. What changes company should bring to reduce the number of absenteeism?

2. How much losses every month can we project in 2011 if same trend of absenteeism continues?

1.2 Data:

Dataset Details:

Dataset Characteristics: Timeseries Multivariant
Number of Attributes: 21
Missing Values : Yes

Attribute Information:

1. Individual identification (ID)
2. Reason for absence (ICD). Absences attested by the International Code of Diseases (ICD) stratified into 21 categories (I to XXI) as follows:
   - I. Certain infectious and parasitic diseases
   - II. Neoplasms
   - III. Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism
   - IV. Endocrine, nutritional and metabolic diseases
   - V. Mental and behavioural disorders VI Diseases of the nervous system VII Diseases of the eye and adnexa
   - VI. Diseases of the ear and mastoid process
   - VII. Diseases of the circulatory system
   - VIII. Diseases of the respiratory system
   - IX. Diseases of the digestive system
   - X. Diseases of the skin and subcutaneous tissue
   - XI. Diseases of the musculoskeletal system and connective tissue
   - XII. Diseases of the genitourinary system
   - XIII. Pregnancy, childbirth and the puerperium
   - XIV. Certain conditions originating in the perinatal period
   - XV. Congenital malformations, deformations and chromosomal abnormalities

XVI. Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
XVII. Injury, poisoning and certain other consequences of external causes
XVIII. External causes of morbidity and mortality
XIX. Factors influencing health status and contact with health services

And 7 categories without (CID) patient follow-up (22), medical consultation (23), blood donation (24), laboratory examination (25), unjustified absence (26), physiotherapy (27), dental consultation (28).

3. Month of absence
4. Day of the week (Monday (2), Tuesday (3), Wednesday (4), Thursday (5), Friday (6))
5. Seasons (summer (1), autumn (2), winter (3), spring (4))
6. Transportation expense
7. Distance from Residence to Work
8. Service time
9. Age
10. Work load Average/day
11. Hit target
12. Disciplinary failure (yes=1; no=0)
13. Education (high school (1), graduate (2), postgraduate (3), master and doctor (4))
14. Son (number of children)
15. Social drinker (yes=1; no=0)
16. Social smoker (yes=1; no=0)
17. Pet (number of pet)
18. Weight
19. Height
20. Body mass index
21. Absenteeism time in hours (target)

2. Methodology

2.1 Pre-Processing

Data preprocessing is a data science technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing.

Data-gathering methods are often loosely controlled, resulting in out-of-range values, impossible data combinations, missing values, etc. This is often called as exploratory data analysis.

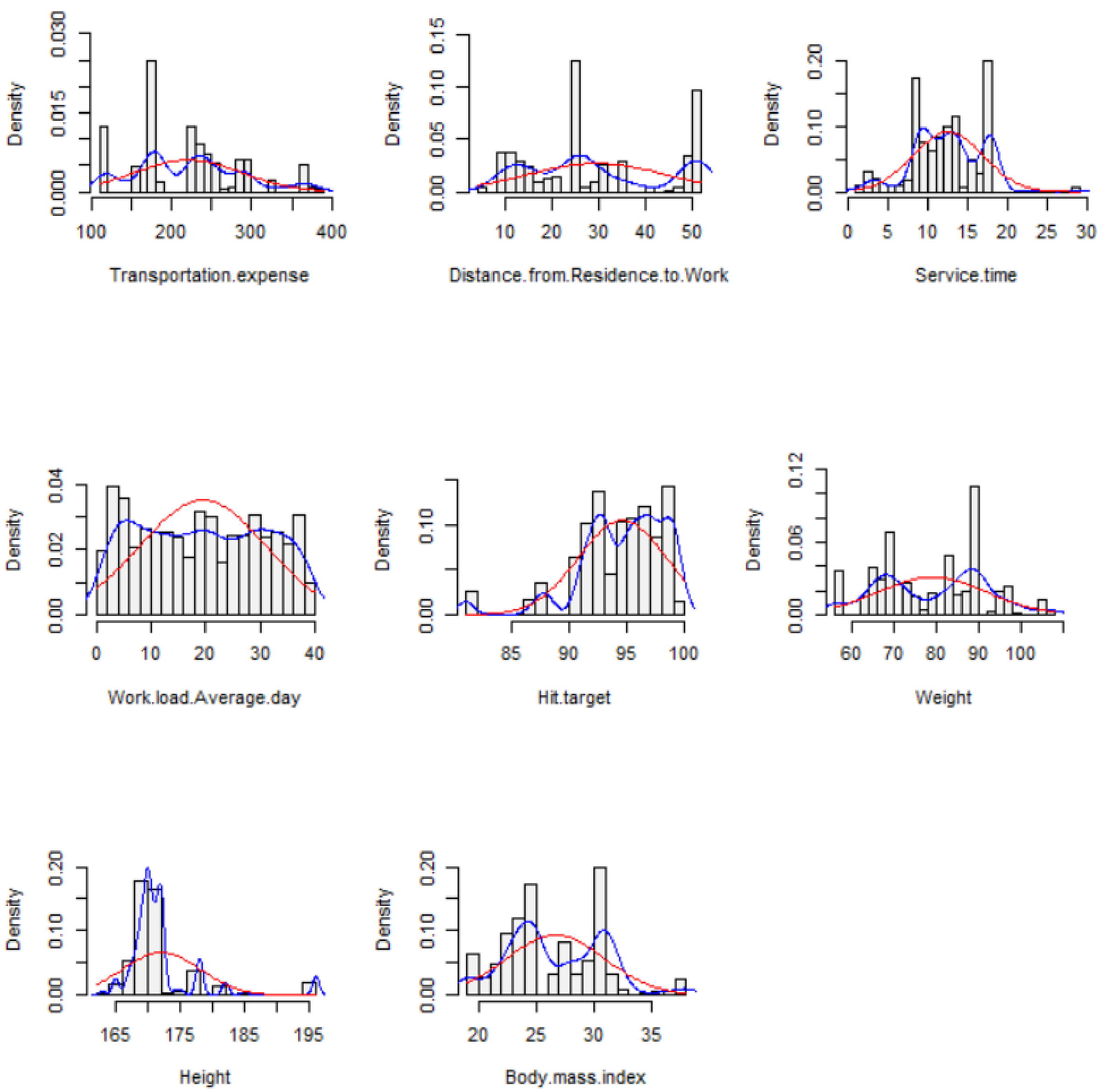
2.2.1 Missing Value Analysis

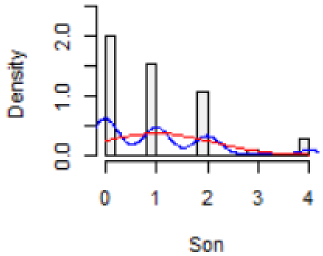
Missing value can arise due to many cases and Proper handling of missing values is important in all statistical analyses. Here Target variable is 'Absenteeism.time.in.hours'. Which itself contains some missing values, in such cases we should neglect the observation. 22 observations from 740 observation has no Target variable. Missing values are imputed using different methods such as Mean, median and KNN imputation. The criterion for imputation is that the variable should have missing values less than 30 percent, In this case no variable has missing values more than 30 percent. To choose the method for imputation we purposefully create a NA and try to impute it using different methods, whichever method gives the closest output, we freeze that method. Method may vary from variable to variable and it mainly depends upon the whether variable is categorical or continuous. Following is the percentage of missing values in each variable

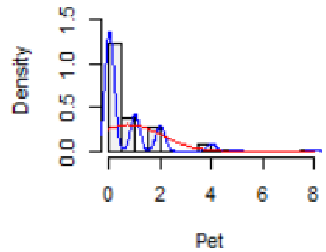
	variable	no_of_missing_value	Missing_percentage
1	ID	0	0.0000000
2	Reason.for.absence	3	0.4054054
3	Month.of.absence	1	0.1351351
4	Day.of.the.week	0	0.0000000
5	Seasons	0	0.0000000
6	Transportation.expense	7	0.9459459
7	Distance.from.Residence.to.Work	3	0.4054054
8	Service.time	3	0.4054054
9	Age	3	0.4054054
10	Work.load.Average.day	0	0.0000000
11	Hit.target	6	0.8108108
12	Disciplinary.failure	6	0.8108108
13	Education	10	1.3513514
14	Son	6	0.8108108
15	Social.drinker	3	0.4054054
16	Social.smoker	4	0.5405405
17	Pet	2	0.2702703
18	Weight	1	0.1351351
19	Height	14	1.8918919
20	Body.mass.index	31	4.1891892
21	Absenteeism.time.in.hours	22	2.9729730

2.2.2 Outlier Analysis

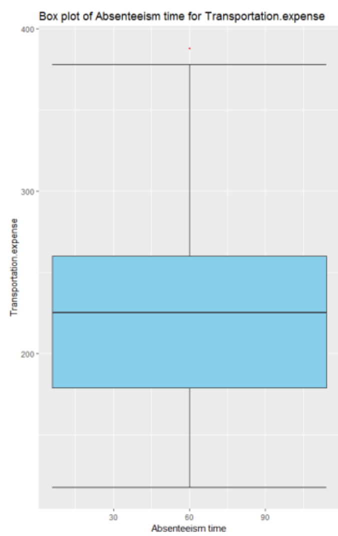
Outliers are extreme values that deviate from other observations on data, they may indicate a variability in a measurement, experimental errors. This data contains outliers in variables such as ‘Transportation Expense’, ‘Service time’, ‘Hit target’ and some personal details. Below are histograms of numerical variables. We can see the distribution is not normal hence we need to perform outlier analysis which will try to impute the outliers with medians or means of the data. Although distribution plots does not help us to find the outliers, we use boxplot method to identify and remove or impute the outliers. We visualize the data using boxplots.

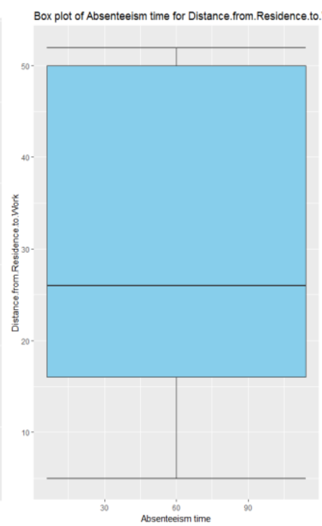


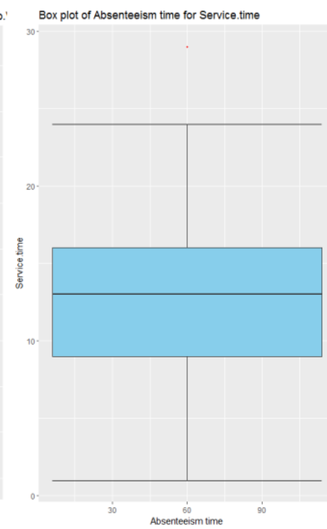


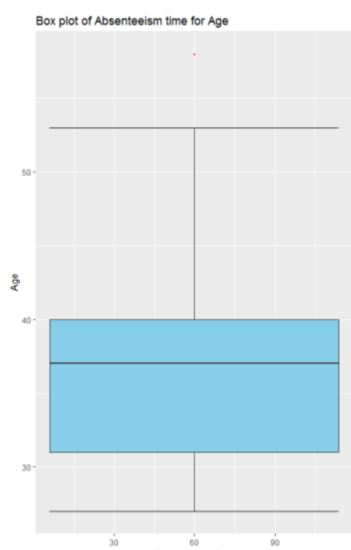


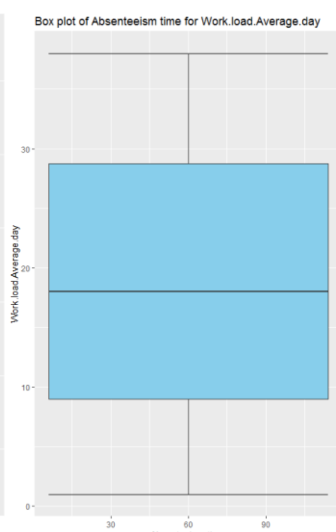
We can clearly see the distribution is not normal hence below are the boxplots for the same. In these boxplots the observations which are above or below 1.5 times the interquartile range are marked as outliers. Interquartile range is shaded as blue while outliers are marked as Red dots.

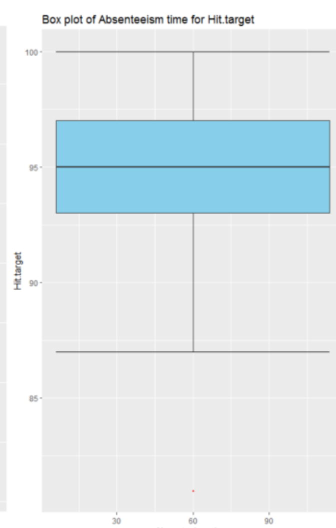




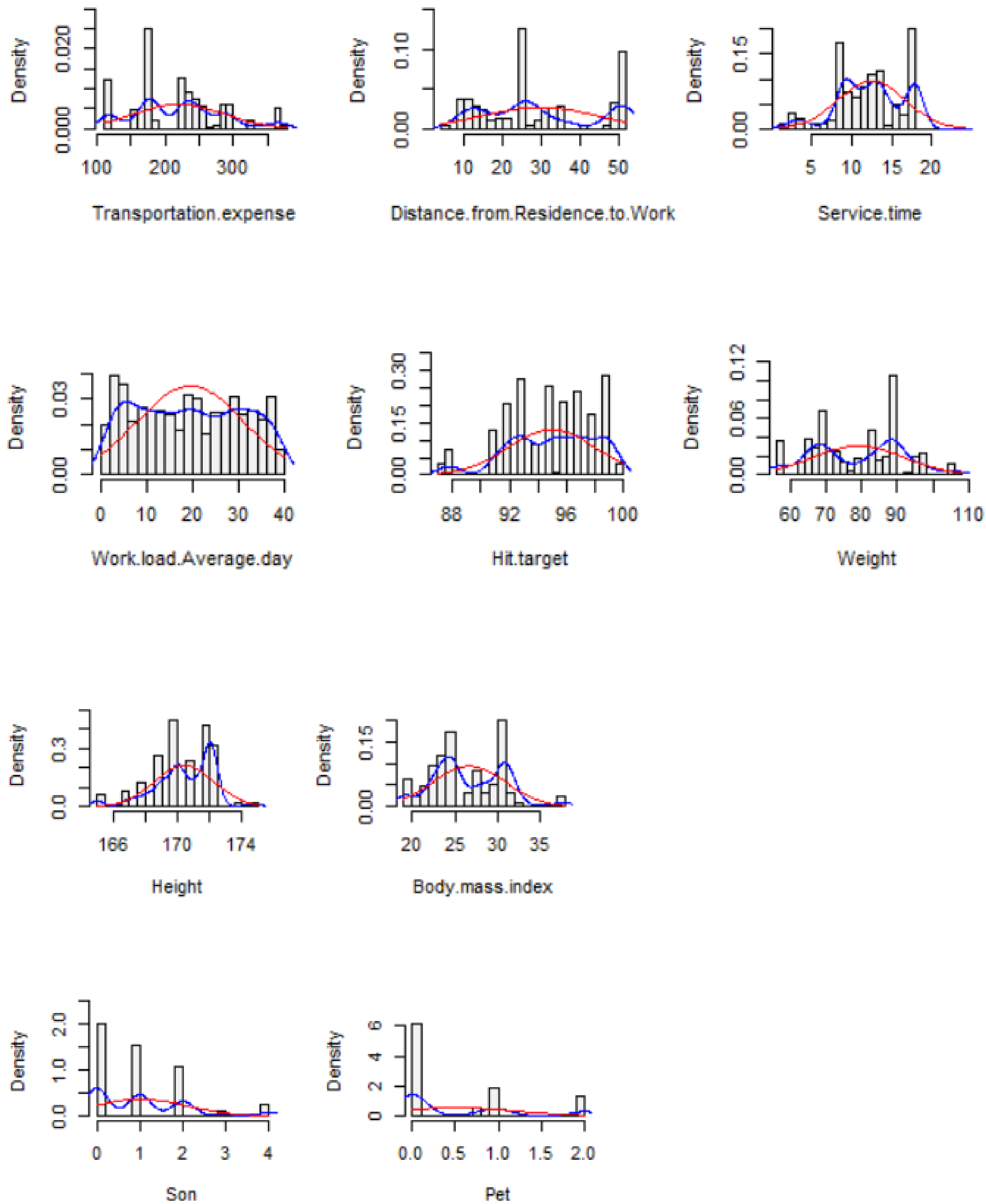








We used the boxplot method to identify and remove the outliers from the variables and the histogram distribution after removal of outliers is given below



2.2.3 Feature Selection

Before performing any type of modeling we need to assess the importance of each predictor variable in our analysis. There is a possibility that many variables in our analysis are not important at all to the problem of class prediction. There are several methods of doing that. We are here using correlation plot to identify the correlation between numerical variables and we will neglect the variable which are highly correlated to each other so that they does not carry the same information to the model development. In same way we will use Chi-Square test for categorical variable.

The chi-square test is a statistical test of independence to determine the dependency of two variables. It shares similarities with coefficient of determination, R^2. However, chi-square test is only applicable to categorical or nominal data while R^2 is only applicable to numeric data. Below, we can see the output of Chi-square test of independence for categorical variables from given data

```
ID
1.4954158388291075e-65
Reason for absence
1.5194795569729106e-136
Month of absence
3.432975009623508e-55
Day of the week
1.1353710371238763e-40
Seasons
2.790172171150468e-52
Education
0.999355371240261
Social drinker
6.958462386179855e-17
Social smoker
0.42109821908319545
Disciplinary failure
2.207076113445463e-120
```

This test resides on P value of the variable. The null hypothesis of this test is that the two variables are independent of each other hence Alternate hypothesis would be that they are dependent on each other hence if the P value is greater than 0.05 which allows null hypothesis to be correct and states that the variables are not dependent on each other.

Here, ‘Education’ and ‘Social smoker’ are two categorical variables which follows null hypothesis and hence the target variable is independent of these two variables. Hence we will drop these two variable and proceed with other for model development

Correlation plot for the numerical variables from data is given below

Here, Dark Red indicates that the two variables are highly positively correlated to each other and Dark Blue indicates that the two variables are highly negatively correlated to each other. As we can see 'Weight' and 'Body mass index' are highly positively correlated to each other so we should drop any one variable from them to avoid multicollinearity.

Using above two techniques to eliminate variables, we have dropped 'Pet', 'Age', 'Son', 'Weight', 'Height','Education','Social smoker' from the data set.

2.2.4 Feature Scaling

Feature scaling is a method used to standardize the range of independent variables or features of data.
There are two methods of feature scaling viz. Normalization and standardization.
Standardization is applied when data is normally distributed and normalization is applied in other cases. Here we have used normalization as there is skewness in some variables

3. Modelling

3.1 Model selection

Model selection depends on the Target variable. In case of given problem statement and dataset the target variable is continuous, hence the model will be regression model.
For regression model there are many models with which we can train our data and test on the same. We will consider some models in here and then depending on error rate we will decide on the same.

3.1.1 Multiple linear Regression

```
#Multiple linear regression
#train, test = train_test_split(df_empabs, test_size=0.2)
#MLR_model = sm.OLS(train.iloc[:,13], train.iloc[:,0:13]).fit()
#MLR_model.summary()
```

Dep. Variable:	Absenteeism time in hours	R-squared:	0.308
Model:	OLS	Adj. R-squared:	0.282
Method:	Least Squares	F-statistic:	11.83
Date:	Sun, 12 Aug 2018	Prob (F-statistic):	3.07e-21
Time:	05:48:19	Log-Likelihood:	-1431.2
No. Observations:	359	AIC:	2888.
Df Residuals:	346	BIC:	2939.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
ID	0.0883	0.079	1.125	0.262	-0.066	0.243
Reason for absence	-0.3298	0.100	-3.288	0.001	-0.527	-0.133
Month of absence	0.3098	0.260	1.189	0.235	-0.203	0.822
Day of the week	-1.0604	0.535	-1.982	0.048	-2.113	-0.008
Seasons	-0.0150	0.774	-0.019	0.985	-1.538	1.508
Transportation expense	10.8724	3.346	3.249	0.001	4.291	17.454
Distance from Residence to Work	-6.0748	2.950	-2.059	0.040	-11.877	-0.272
Service time	13.9477	4.973	2.805	0.005	4.167	23.729
Work load Average/day	2.0219	3.043	0.664	0.507	-3.963	8.007
Hit target	8.1132	3.099	2.618	0.009	2.019	14.208
Disciplinary failure	-15.0622	4.141	-3.637	0.000	-23.207	-6.917
Social drinker	4.0780	1.857	2.197	0.029	0.426	7.730
Body mass index	-3.7895	3.686	-1.028	0.305	-11.040	3.461

Omnibus:	399.438	Durbin-Watson:	1.817
Prob(Omnibus):	0.000	Jarque-Bera (JB):	17520.370
Skew:	5.028	Prob(JB):	0.00

As you can see the Adjusted R-squared value, we can explain only about 30% of the data using our multiple linear regression model. This is not very impressive, but at least looking at the F-statistic and combined p-value we can reject the null hypothesis that target variable does not depend on any of the predictor variables.
After changing the test data to 40 percent the model gave output as follow

```
#Multiple linear regression
#train, test = train_test_split(df_empabs, test_size=0.4)
#MLR_model = sm.OLS(train.iloc[:,13], train.iloc[:,0:13]).fit()
#MLR_model.summary()
```

Dep. Variable:	Absenteeism time in hours	R-squared:	0.307
Model:	OLS	Adj. R-squared:	0.290
Method:	Least Squares	F-statistic:	17.90
Date:	Sun, 12 Aug 2018	Prob (F-statistic):	1.75e-34
Time:	07:32:51	Log-Likelihood:	-2092.3
No. Observations:	538	AIC:	4211.
Df Residuals:	525	BIC:	4266.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
ID	0.1161	0.055	2.115	0.035	0.008	0.224
Reason for absence	-0.3420	0.073	-4.696	0.000	-0.485	-0.199
Month of absence	0.6420	0.185	3.467	0.001	0.278	1.006
Day of the week	-0.7277	0.378	-1.923	0.055	-1.471	0.016
Seasons	-0.9000	0.568	-1.584	0.114	-2.016	0.216
Transportation expense	6.9846	2.402	2.908	0.004	2.266	11.703
Distance from Residence to Work	-3.2168	2.107	-1.527	0.127	-7.355	0.922
Service time	9.6767	3.704	2.612	0.009	2.400	16.953
Work load Average/day	4.8111	2.265	2.124	0.034	0.361	9.261
Hit target	6.5114	2.250	2.894	0.004	2.092	10.931
Disciplinary failure	-13.4982	2.786	-4.845	0.000	-18.971	-8.025
Social drinker	3.2906	1.374	2.395	0.017	0.591	5.990
Body mass index	-2.3357	2.688	-0.869	0.385	-7.617	2.945

Omnibus:	625.225	Durbin-Watson:	2.117
Prob(Omnibus):	0.000	Jarque-Bera (JB):	40425.600
Skew:	5.596	Prob(JB):	0.00
Kurtosis:	43.965	Cond. No.	239.

```
MLR_model.summary()

# Dep. Variable:
# Absenteeism time in hours
# R-squared: for testsize=0.2
# 0.307
# Adj. R-squared:
# 0.289

# Dep. Variable:
# Absenteeism time in hours
# R-squared: for testsize=0.3
# 0.293
# Adj. R-squared:
# 0.272

# Dep. Variable:
# Absenteeism time in hours
# R-squared: for testsize=0.4
# 0.318
# Adj. R-squared:
# 0.295

# Dep. Variable:
# Absenteeism time in hours
# R-squared: for testsize=0.5
# 0.330
# Adj. R-squared:
# 0.301
```

After changing the test data also it did not change the predictive power of our regression model effectively. Therefore, this is the maximum accuracy that we can get from this model

```
#Calculate MAE
MAE(test.iloc[:,13], predictions_MLR)
```

#MAE 5.9 for test size=0.3
#MAE 5.8 for test size=0.2
#MAE 5.7 for test size=0.4
#MAE 6.34 for test size=0.5
#MAE 6.83 for test size=0.15

3.1.2 KNN Regressor

```
train, test = train_test_split(df_empabs, test_size=0.2)
KNN_model = KNeighborsRegressor(n_neighbors=3).fit(train.iloc[:,0:13], train.iloc[:,13])
predictions_KNN = KNN_model.predict(test.iloc[:,0:13])
#Calculate MAE
MAE(test.iloc[:,13], predictions_KNN)
```

#for KNN=3
#MAE 4.19 for test size=0.3
#MAE 3.63 for test size=0.2
#MAE 4.39 for test size=0.4
#MAE 4.14 for test size=0.15

#for KNN=2
#MAE 4.20 for test size=0.3
#MAE 4.53 for test size=0.2
#MAE 3.79 for test size=0.4
#MAE 5.33 for test size=0.15

3.1.3 Random Forest Regressor

```
train, test = train_test_split(df_empabs, test_size=0.2)
RFR_model = RandomForestRegressor(n_estimators = 20).fit(train.iloc[:,0:13], train.iloc[:,13])
RFR_Predictions = RFR_model.predict(test.iloc[:,0:13])
#Calculate MAE
MAE(test.iloc[:,13], RFR_Predictions)
```

#MAE 5.86 for test size=0.2
#MAE 5.2 for test size=0.3

4. Conclusion

4.1 Model Evaluation

Now that we have a few models for predicting the target variable, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare models using any of the following criteria:

1. Predictive Performance
2. Interpretability
3. Computational Efficiency

In this case, if we consider Predictive performance of all models then after comparing their MAE (Mean absolute error) we can pick any one model as the MAE for both models are nearly same. This data set contains very less number of observations which affects the model building and predictions hence with high number of observations we might be able to perform well with the same models. The levels in Target variable were also very wide which resulted in average model building, with high number of observations this problem can also be solved.

4.2 What changes company should bring to reduce the number of absenteeism?

As the requirement of problem statement, we can use Random Forest model to predict the contribution of each independent variable resulted in variance of Target variable

```
> RF_model = randomForest(Absenteeism.time.in.hours ~ ., df_empabs,
                                           importance = TRUE)
> importance(RF_model)
                              %IncMSE IncNodePurity
ID                          4.1255159     7356.1379
Reason.for.absence         12.0008526    23818.3700
Month.of.absence            3.3613963     8883.9971
Day.of.the.week            -0.1893287     7535.9078
Seasons                     4.8723884     4377.5246
Transportation.expense      3.6879140     6915.3141
Distance.from.Residence.to.Work 4.5476416  9642.9045
Service.time                5.1999099     4635.8810
Work.load.Average.day       2.8401655    12388.5741
Hit.target                  0.3666597     6717.5765
Disciplinary.failure       -0.8092790      922.6676
Social.drinker              3.4668085     1602.1034
Body.mass.index             5.6755639     5468.1417
```

As we can observe here, Reason.for.absence contributes most than other variables while it comes to node splitting Reason.for.absence as well as Work.load.Average.day contributes more than other variables hence we can treat these two variables as important.

On Analyzing further for Reason.for.absence, we can aggregate the results with reason codes to related Absenteeism hours which gives result as follows

```
> aggregate(data=df_empabs1,predicted_abs_hrs~Reason.for.absence,sum)
   Reason.for.absence predicted_abs_hrs
1                   1         278.08844
2                   2          15.92431
3                   3          17.75265
4                   4          26.11160
5                   5          39.40333
6                   6         106.36139
7                   7         216.68603
8                   8          72.08159
9                   9          57.12723
10                 10         294.06592
11                 11         276.39156
12                 12          95.07083
13                 13         572.26403
14                 14         189.26185
15                 15          23.18990
16                 16          28.76178
17                 17          10.00768
18                 18         197.27571
19                 19         325.53114
20                 20         212.01087
21                 21          44.95017
22                 22         232.85517
23                 23         801.70911
24                 24          20.02198
25                 25         169.53102
26                 26         162.81082
27                 27         141.98287
28                 28         396.26621
```

Here, we can observe that Reason.for.absence code 13 , 23 are more often responsible for absenteeism. Which are as follows

13 - Pregnancy, childbirth and the puerperium
23 – Medical Consultation

Hence, company should look into these two reasons and workload average per day should be reduced to decrease the rate of absenteeism

4.3 How much losses every month can we project in 2011 if same trend of absenteeism continues?

We can answer this question by feeding whole independent variable's observation to our model and compare the resulted absenteeism hours to the month of absence, which will give us the pattern of monthly absenteeism for further coming year (provided same trend continues)

```
> predictions_LR = predict(lm_model, df_empabs[,1:13])
> df_empabs1$predicted_abs_hrs=predictions_LR
> aggregate(data=df_empabs1, predicted_abs_hrs~Month.of.absence,sum)

   Month.of.absence predicted_abs_hrs
1                 1          227.1016
2                 2          393.3676
3                 3          633.4353
4                 4          335.6969
5                 5          566.5677
6                 6          371.7813
7                 7          574.0996
8                 8          358.5721
9                 9          320.9492
10               10          444.2815
11               11          455.6757
12               12          341.9666
```

These many number of hours are predicted by the model if the same trend continues for upcoming year

5. Appendix

5.1 R Code

```
rm(list=ls())
x = c("ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced", "C50",
"dummies", "e1071", "Information",
     "MASS", "rpart", "gbm", "ROSE", 'sampling', 'DataCombine', 'inTrees',"usdm","class")
lapply(x, require, character.only = TRUE)

df_empabs=read.csv("Absenteeism_at_work_Project.xls")
sum(is.na(df_empabs))

#first remove the observations for which target variable is null
df_empabs = df_empabs[which(!df_empabs$Absenteeism.time.in.hours %in% NA),]
dim(df_empabs)


#data preprocessing
df_empabs$Work.load.Average.day=as.numeric(df_empabs$Work.load.Average.day)
#Change Reason code 0 to 20
df_empabs$Reason.for.absence[which(df_empabs$Reason.for.absence %in% 0)] = 20
#for month = 0 , make it 12
df_empabs$Month.of.absence[which(df_empabs$Month.of.absence %in% 0)] = 12

#missing value analysis
df_empabs$Work.load.Average.day[is.na(df_empabs$Work.load.Average.daye)] =
median(df_empabs$Work.load.Average.day, na.rm = T)
df_empabs$Month.of.absence[is.na(df_empabs$Month.of.absencee)] =
median(df_empabs$Month.of.absence, na.rm = T)
df_empabs$Reason.for.absence[is.na(df_empabs$Reason.for.absence)] =
median(df_empabs$Reason.for.absence, na.rm = T)

#other variables missing values imputation
df_empabs=knnImputation(df_empabs,k=3)


# df=subset(df_empabs,select=
c(Transportation.expense,Distance.from.Residence.to.Work,Service.time,Work.load.Average.da
y,Hit.target,
 #                       Weight,Height,Body.mass.index))
```

```
#outlier analysis
multi.hist(df, main = NA, dcol = c("blue", "red"),
          dlty = c("solid", "solid"), bcol = "grey95")

#boxplot analysis
numeric_var=c("Transportation.expense","Distance.from.Residence.to.Work","Service.time","A
ge","Work.load.Average.day","Hit.target",
           "Son","Pet","Weight","Height","Body.mass.index")


for (i in 1:length(numeric_var))
{
  assign(paste0("gn",i), ggplot(aes_string(y = (numeric_var[i]), x = "Absenteeism.time.in.hours"),
data = df_empabs)+
         stat_boxplot(geom = "errorbar", width = 0.5) +
         geom_boxplot(outlier.colour="red", fill = "skyblue" ,outlier.shape=20,
                   outlier.size=1, notch=FALSE) +
         theme(legend.position="bottom")+
         labs(y=numeric_var[i],x="Absenteeism time")+
         ggtitle(paste("Box plot of Absenteeism time for",numeric_var[i])))
}

#gn1

#boxplot analysis
boxplot.stats(df_empabs$Transportation.expense)$out
val = df_empabs$Transportation.expense[df_empabs$Transportation.expense %in%
boxplot.stats(df_empabs$Transportation.expense)$out]
df_empabs$Transportation.expense[df_empabs$Transportation.expense %in% val] =
mean(df_empabs$Transportation.expense, na.rm = T)
boxplot.stats(df_empabs$Hit.target)$out
val = df_empabs$Hit.target[df_empabs$Hit.target %in%
boxplot.stats(df_empabs$Hit.target)$out]
df_empabs$Hit.target[df_empabs$Hit.target %in% val] = mean(df_empabs$Hit.target, na.rm =
T)
boxplot.stats(df_empabs$Service.time)$out
val = df_empabs$Service.time[df_empabs$Service.time %in%
boxplot.stats(df_empabs$Service.time)$out]
df_empabs$Service.time[df_empabs$Service.time %in% val] = mean(df_empabs$Service.time,
na.rm = T)
boxplot.stats(df_empabs$Age)$out
val = df_empabs$Age[df_empabs$Age %in% boxplot.stats(df_empabs$Age)$out]
df_empabs$Age[df_empabs$Age %in% val] = mean(df_empabs$Age, na.rm = T)
boxplot.stats(df_empabs$Work.load.Average.day)$out
```

```
val = df_empabs$Work.load.Average.day[df_empabs$Work.load.Average.day %in%
boxplot.stats(df_empabs$Work.load.Average.day)$out]
df_empabs$Work.load.Average.day[df_empabs$Work.load.Average.day %in% val] =
mean(df_empabs$Work.load.Average.day, na.rm = T)


#Feature selection
numeric_index=c("Transportation.expense","Distance.from.Residence.to.Work","Service.time","
Age","Work.load.Average.day","Hit.target","Son","Pet","Weight","Height","Body.mass.index")


corrgram(df_empabs[,numeric_index], order = F,
        upper.panel=panel.pie, text.panel=panel.txt, main = "Correlation Plot")

## Chi-squared Test of Independence
factor_index = c("ID","Reason.for.absence", "Month.of.absence"
,"Day.of.the.week","Seasons","Disciplinary.failure","Education","Social.drinker","Social.smoker"
)
factor_data = df_empabs[,factor_index]

for (i in 1:9)
{
  print(names(factor_data)[i])
  print(chisq.test(table(df_empabs$Absenteeism.time.in.hours,factor_data[,i]),simulate.p.value
= TRUE))
}

df_empabs = subset(df_empabs,select = -
c(Age,Son,Pet,Weight,Height,Education,Social.smoker))

#feature scaling

#Normalisation
cnames = c("Transportation.expense"
,"Distance.from.Residence.to.Work","Service.time","Work.load.Average.day","Hit.target","Body
.mass.index" )

for(i in cnames){
  print(i)
  df_empabs[,i] = (df_empabs[,i] - min(df_empabs[,i]))/
    (max(df_empabs[,i] - min(df_empabs[,i])))
}
```

```
#sampling
train_index = sample(1:nrow(df_empabs), 0.8 * nrow(df_empabs))
train = df_empabs[ train_index,]
test  = df_empabs[-train_index,]

#Linear Regression
vif(df_empabs[,-14])
lm_model = lm(Absenteeism.time.in.hours ~., data = train)
summary(lm_model)
predictions_LR = predict(lm_model, test[,1:13])

#MAE
MAE = function(y, yhat){
  mean(abs((y - yhat)))
}
#Calculate MAE
MAE(test[,14], predictions_LR)

#KNN regressor model
k= knn(train[,1:13],test[,1:13],train$Absenteeism.time.in.hours, k=3)
MAE(test[,14], as.numeric(k))

library("randomForest")
RF_model = randomForest(Absenteeism.time.in.hours ~ ., train, importance = TRUE)
importance(RF_model)
predictions_RF = predict(RF_model, test[,1:13])
MAE(test[,14], predictions_RF)


#Predict for problem statement
predictions_LR = predict(lm_model, df_empabs[,1:13])
df_empabs1=df_empabs
df_empabs1$predicted_abs_hrs=predictions_LR
aggregate(data=df_empabs1,predicted_abs_hrs~Reason.for.absence,sum)
aggregate(data=df_empabs1,predicted_abs_hrs~Month.of.absence,sum)
```