Predicting Employee Absenteeism using Machine Learning - A classification approach

Name: Sheraz Ahmed
Module: Advance Analytics and Machine Learning
Semester: 2
Program: MSc (T) Business Analytics

Abstract:

Predicting employee absenteeism has been studied for a long time, however most research focus on factor impacting absenteeism. This study divided the absenteeism between normal and abnormal and analysed the factors leading to abnormal absenteeism, which gave the study a different dimension. This study analysed the employee absenteeism dataset of a Brazilian Courier Company using the classification approach. The machine learning algorithms used in the study were LASSO Logistic Regression, XGboost, Random Forest, Support Vector Machines and Decision Tree. 10-fold cross validation approach was used for all algorithms along with grid search for parameter tuning. Our findings indicate that Decision Tree is the best model for prediction of abnormal absences with accuracy of 82.31% and Kappa value of 0.6391. LASSO Logistic regression also yielded accuracy of 82.31% but slightly lower of kappa value of 0.6224. XGboost yielded accuracy of 82.31% with Kappa value of 0.6195. Random forest and SVM yielded lowest accuracy of 80.95% with kappa value of 0.5934 and. 0.5995 respectively.

1. Introduction:

Absenteeism can be defined as a habitual absence from work for a day or more than one day (Cucchiella et al., 2014). Employees are considered as human capital of a firm as its success depends on the employees' commitment (Ali Shah et al., 2020). Increased absenteeism can lead to increased cost for an organization and can act as a barrier in achieving its objectives and goals (Dogruyo & Sekeroglu, 2019). Increased absenteeism is also related to decreased productivity of the employees (Wahid et al., 2019). Some researchers have also related increased employee absenteeism with decreasing purchasing power of the employees along with increased psychological burden (Dogruyo & Sekeroglu, 2019). Nanjundeswaraswamy (2016) notes that reduction in employee absenteeism also shows a positive effect on gross domestic product. The impact of absenteeism can be gauged by the fact that bureau of labour statistics reported that 2.8 million workdays are lost due to employee absenteeism (Truman, 2003). Haswell (2003) noted that abnormal, long-term absences result in the loss of 40% of the total working time. Kocakulah et al. (2016) noted that total cost of unplanned absences accounts for 20% of their total payroll expense. It was noted that unplanned and abnormal absences were costing a major airline company approximately $1 million per day (Kaleta & Edward, 2003)

Cause of employee absenteeism:

Employee absenteeism is caused to various factors, some of them are noted by Kocakulah et al. (2016). They noted that in 2002, illness accounted for 33% of the unscheduled absences, while 24% absences were due to family problems, 21% due to personal needs, 12% due to stress and 10% due to employees' sense of entitlement (Kocakulah et al., 2016; Truman, 2003). In the United States of America, work related stress and lack of work life balance has resulted in cost of $200 for the industry (Kocakulah et al., 2016, p. 91). Employee absenteeism is considered a problem of Human Resource Management generally (Bycio, 1992). It is considered one of the strategic objectives of Human Resource Management to ensure organizational growth (Halbesleben et al., 2014).

Prior Literature:

Various authors have used machine learning techniques to predict employee absentees. In this section we will discuss the prior literature relating to the use of classification methods for predicting employee absenteeism.

Oliveira et al (2019) analysed the data set of employees of a call centre of a major Brazilian telecommunication company. They analysed the population of 13,805 employees and 241 attributes of employees. The methods used by Oliveira et al. (2019) included different classification algorithms such as Random Forest, Support Vector Machines, Naïve Bayes, XGBoost, Multilayer Perceptron and Long Short Term Memory. They used evolutionary algorithms for tuning of parameters and found that XGBoost yielded the predictive accuracy of 72% while Random Forest yielded a predictive accuracy of 71%.

Ali Shah et al. (2020) studied the same dataset from a Brazilian Courier Company as we have used in this research. They applied Deep Neural Network, Single Neural Network, Support Vector Machines, Decision Tree, and Random Forest algorithms to train and test the predictive accuracy of the model. They found the highest accuracy of 90.6% using Deep Neural Network while Single Neural Network resulted in73.3%. The Decision Tree, Support Vector Machine and Random Forest algorithms yielded predictive performance of 82%.

Dogruyol & Sekeroglu (2019) used various neural networks such as Long-Short Term Memory Neural Network, Backpropagation Method-Based Neural Network and Radial-Basis Function-Based Neural Network. They found that Long-Short Term Memory Neural Network yielded 99% predictive accuracy.

Wahid et al. (2019) analysed the same data set from a Brazilian Courier Company and applied Gradient Boosted Trees, Random Forest, Tree Ensemble and Decision Tree algorithms. They used 7 evaluation metrics such as Sensitivity, Specificity, Accuracy, True Positive, True Negative, False Positive and False Negative to measure the predictive performance of the algorithms. They reported predictive accuracy of 82% using Gradient Boosted Trees and 79% with Tree Ensemble.

Skorikov et al. (2020) analysed the same data set as this research but divided the outcome variable in three categories i.e., class A for for 0 hours of absent, class B for 1-15 hours of absence and 16-120 for class C. They analysed the dataset using zeroR, tree-based J48, K nearest Neighbour and naïve Bayes classifier algorithms using 10 fold cross validation technique. They reported the predictive performance of 90.9% for KNN, 90.1% for naïve Bayes and 89.1% for J48.

Nath et al. (2022) also analysed the same dataset from Brazilian Courier Company with aim of predicting employee absenteeism while also developing a web-based interactive tool for HR managers to use to predict absentees without applying complex machine learning algorithms themselves. They applied Multinomial Logistic Regresion, Support Vector Machines, Artificial Neural Networks and Random Forest to analyse the dataset (Nath et al., 2022). They employed Accuracy, Precision, Recall, F1 Score and ROC AUC score as performance measures (Nath et al., 2022). Multinomial Logistic Regression yielded the highest accuracy of 0.932, while Support Vector Machine yielded the predictive accuracy score of 0.887 (Nath et al., 2022). Artificial Neural Network resulted in predictive accuracy score of 0.873 and Random Forest yielded the lowest predictive accuracy of 0.869.

Research Objective:

Employee absenteeism is seen as an indicator of decreasing employee engagement and commitment and also emphasises the need for the employer to take adequate actions to tackle it (Cohen & Golan, 2007). In the recent years, the focus shifted towards predicting employee absenteeism in advance using machine learning techniques so that an organization can have an idea of factors impacting abnormal absences and take adequate measures against it. In this research, we have used the absenteeism dataset from a Brazilian Courier company, originally used by Martiniano et al. (2012) and obtained through UCI Machine Learning Repository. This data set has also been used by Wahid et al. (2019) Ajmi (2020), Ali Shah et al (2020), Skorikov et al (2020) and Nath et al. (2022). For the purpose of this research, we classify the outcome variable i.e. absentees in two categories i.e., “Normal” and “Abnormal”. Normal Absentees were those that were shorted than 6 hours in length and abnormal absentees were those that were greater than 6 hours in length. As noted by Ali Shah et al. (2020) that generally working hours in an organization are eight hours/day. The employee being absent for more than 50% of working hours is considered abnormal. Ajmi (2020) also classified the long and short absent in the similar manner. Ali Shah et al. (2020), who used the same dataset classified abnormal absence as more than 5 hours. In this research, we have used various machine learning algorithms with different tuning parameters to predict the probability of abnormal absentees. The algorithms used are as follow:

1. LASSO Logistic Regression
2. Support Vector Machine
3. Random Forest
4. XG Boost
5. Decision Tree

The predictive accuracy of each algorithm was measured using Accuracy, Kappa Value and ROC as the performance metrics. The model with highest Accuracy was considered the best model. The objective of this research is to discover the algorithm that yields the highest predictive accuracy for absenteeism prediction.

2. Methodology:

Data set Information:

As previously mentioned, the data set used for this research was obtained through UCI Machine Learning Repository (UCI, no date). The dataset was collected from a courier company in Brazil. It consists of records of employee absences from the period of July 2007 to July 2010. The dataset consists of 21 attribute and 740 observations (UCI). The dataset was initially used by Martiniano et al., (2012) but has since been used by various researcher as mentioned in the literature review section above. The dataset contains information relating to workload, health, habits, traveling and various other attributes (UCI, no date; Martiniano et al., 2012; Skorikov et al., 2020) . For detailed description of the data set refer to appendix 1. As per UCI (no date), the dataset allows for several manipulation and combinations of new attributes. Various researchers have used the same dataset for classification problems such as Nath et al (2022), Shorikov et al. (2020), Ali Shah et al (2020) and Wahid et al (2019).

Data pre-processing:

Initially the raw dataset was load into the R Studio, the software used to analyse the data. Initial descriptive statistics were analysed using the summary function in R. Suitable actions were taken to pre-process the data into a format suitable for the objective of the research. Initially we renamed all the column names, so they do not contain any spaces. Afterwards, various attributes were converted to type factor. The code used to convert the variables to type factor is provided below.

```
#convert factor variables to type factor

data$reason_for_absence <- as.factor(data$reason_for_absence)
data$month_of_absence <- as.factor(data$month_of_absence)
data$day_of_week <- as.factor(data$day_of_week)
data$Seasons <- as.factor(data$Seasons)
data$disciplinary_fail <- as.factor(data$disciplinary_fail)
data$Education <- as.factor(data$Education)
data$social_drinker <- as.factor(data$social_drinker)
data$social_smoker <- as.factor(data$social_smoker)
data$Pet <- as.factor(data$Pet) #coding number of  pets as a factor as well
```

Figure 1 Convert factor variables to type factor in R

As can be seen from figure 1, we converted reason for absence, month of absence, day of week, seasons, disciplinary failure, education, social drinker, social smoker, and pet variables to a type of factor.

Afterwards, we ran a summary function on the outcome variable that is absenteeism in hours and discovered that mean time of absence was 6.924. We noted that researchers such as Ali Shah et al. (2020) had classified absences longer than 5 hours as abnormal and Ajmi (2020) used 6 hours as the limit for normal absences. We analysed the mean of the outcome variable and found that it was 6.924. However, to avoid imbalance In the dataset, we classified the Normal Absenteeism as lower than 6 hours and Abnormal as greater than 6 hours. These two categories were made part of a new outcome variable called “absent_type” This is consistent with the fact that employees absent for more than 50% of the working day should be considered as abnormal absentees (Ali Shah et al., 2020). The absenteeism_time variable was dropped from the dataset as it would impact the results of the models.

Last but not the least, missing values test was performed, and it was noted that the dataset had no missing values. The screenshot of dataset from the environment is shown below:

```
modeldata                      740 obs. of 20 variables
  $ reason_for_absence      : Factor w/ 28 levels "0","1","2","3",..: 26 1 23 8 23 23 22…
  $ month_of_absence        : Factor w/ 13 levels "0","1","2","3",..: 8 8 8 8 8 8 8 8 8 …
  $ day_of_week             : Factor w/ 5 levels "2","3","4 $ month_of_absence  5 5 5 1 1…
  $ Seasons                 : Factor w/ 4 levels "1","2","3 : Factor w/ 13 levels 1 1 1 1 ...
                                                           "0","1","2","3",..: 8 8 8
  $ transport_expense       : num [1:740] 289 118 179 279 2 8 8 8 8 8 8 8 … 235 ...
  $ distance_residence_work: num [1:740] 36 13 51 5 36 51 52 50 12 11 ...
  $ service_time            : num [1:740] 13 18 18 14 13 18 3 11 14 14 ...
  $ Age                     : num [1:740] 33 50 38 39 33 38 28 36 34 37 ...
  $ avg_workload_day        : num [1:740] 239554 239554 239554 239554 239554 ...
  $ hit_target              : num [1:740] 97 97 97 97 97 97 97 97 97 97 ...
  $ disciplinary_fail       : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...
  $ Education               : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 3 ...
  $ Son                     : Factor w/ 5 levels "0","1","2","3",..: 3 2 1 3 3 1 2 5 3 2…
  $ social_drinker          : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 1 ...
  $ social_smoker           : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
  $ Pet                     : Factor w/ 6 levels "0","1","2","4",..: 2 1 1 1 2 1 4 1 1 2…
  $ Weight                  : num [1:740] 90 98 89 68 90 89 80 65 95 88 ...
  $ Height                  : num [1:740] 172 178 170 168 172 170 172 168 196 172 ...
  $ BMI                     : num [1:740] 30 31 31 24 30 31 27 23 25 29 ...
  $ absent_type             : Factor w/ 2 levels "Abnormal","Normal": 2 2 2 2 2 2 1 2 1 …
```

Figure 2 Dataset configuration in R

The final dataset used for model was split between training and test set. The training set consisted of 80% of the observations while the test set consisted of the remaining 20%. The training set was used to train the models while the test set was used to test the predictive accuracy of the respective models.

Machine Learning Algorithms and tuning parameters used:

We used LASSO logistic regression, Random Forest, Support Vector Machines, XGboost and Decision Tree Algorithms to analyse the dataset and measure the predictive accuracy. 10-Fold Cross Validation approach was used for all the models. In this section we will discuss the mechanism of each algorithm used in this study.

1. LASSO Logistic Regression

Logistic Regression is a generalised linear model that is used to predict the outcome of binary variable or multiple outcomes (in case of non-binary outcome). It calculates the probability of an observation falling under the certain class (0,1), in our case its normal or abormal respectively. Instead of modelling the response directly, as is the case with multiplie linear regression, the logistic regression computes the probabilities of an observation falling under a particular class (Wang & Zhou, 2015). The logistic function is presented as follows:

$$\log\frac{\text{Prob}(Y=1|x)}{\text{Prob}(Y=0|x)} = \beta_0 + \mathrm{x}^T\beta$$

Source: Wang & Zhou, 2015

Where β0 denotes the intercept, β = (β1, ⋯, βp) represents the linear coefficients and Prob Y =1| x and Prob Y =0| x represents the conditional probabilities of class label 0 and 1, or normal and abnormal in our case. Maximum likelihood approach is generally used to calculate the values of coefficients and log-likelihood is represented as:

$$\begin{aligned} l(\beta_0,\beta) &= \sum_{i=1}^{n}\Big\{y_i log Prob(Y=1;\beta) \\ &\quad +(1-y_i)log(1-Prob(Y=1;\beta))\Big\} \\ &= \sum_{i=1}^{n}\Big\{y_i(\beta_0+x_i^T\beta) \\ &\quad -log(1+e^{\beta_0+x_i^T\beta})\Big\} \end{aligned}$$

Source: Wang & Zhou, 2015

LASSO logistic regression uses a penalty regularization method called L1, which uses a tuning parameter known as lambda (James et al., 2013). Sufficient lambda value to calculate coefficients equal to zero (James et al., 2013). The value for sufficient lambda is calculated using 10-fold cross validation approach in our model to select the best model LASSO logistic regression model is represented as follow:

$$\sum_{i=1}^{n}\Big\{log(1+e^{\beta_0+x_i^T\beta})-y_i(\beta_0+x_i^T\beta)\Big\}+\lambda\sum_{j=1}^{p}|\beta_j|$$

Source: Wang & Zhou, 2015

LASSO logistic regression is helpful in feature selection from datasets with high dimensions. The logistic regression algorithm has been used by various researchers for classification. Some of them include Meier et al. (2008) and Wang & Zhou (2015). We used ROC as the performance metric in caret package of R. Confusion Matrix was then made for the model to measure accuracy and kappa value.

2. Random Forest:

Random forest is similar to decision trees, it works by constructing multiple decision tree model and each model is trained based on the different set of attributes and observation (Wahid et al., 2019). It is composed of trees that produce class prediction for each tree and choses the model prediction based on class with most votes (Sarica et al., 2017). For the best random forest model, we used repeated cross validation with 10 folds. We used grid search method as the tuning parameter method for the trees, the value of mtry was derived using random search Random Forest model and the best mtry value was discovered as 29 in the random search. For grid search, we set the values between 1 to 30 for the final model. Ramadhan et al. (2017) has used grid search for tuning the random forest model.

3. Support Vector Machines:

Support vector machines are an extension fo support vector classifiers that that are yielded from enlarging the feature space using a kernels (James et al., 2013). A hyperplane capable of distinguishing between two classes of data is prepared in Support Vector Machines (Nath et al., 2022).

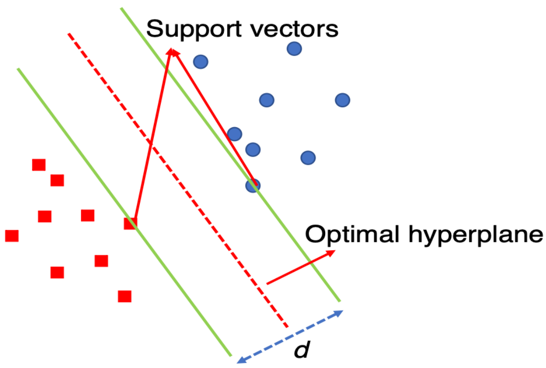


Figure 2. Optimal classification algorithm

The linear support vector can be classified as follow:

- The linear support vector classifier can be represented as

$$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle, \qquad (9.18)$$

  where there are n parameters α_i, $i = 1, \ldots, n$, one per training observation.

- To estimate the parameters $\alpha_1, \ldots, \alpha_n$ and β_0, all we need are the $\binom{n}{2}$ inner products $\langle x_i, x_{i'} \rangle$ between all pairs of training observations. (The notation $\binom{n}{2}$ means $n(n-1)/2$, and gives the number of pairs among a set of n items.)

Source: James et al., 2013

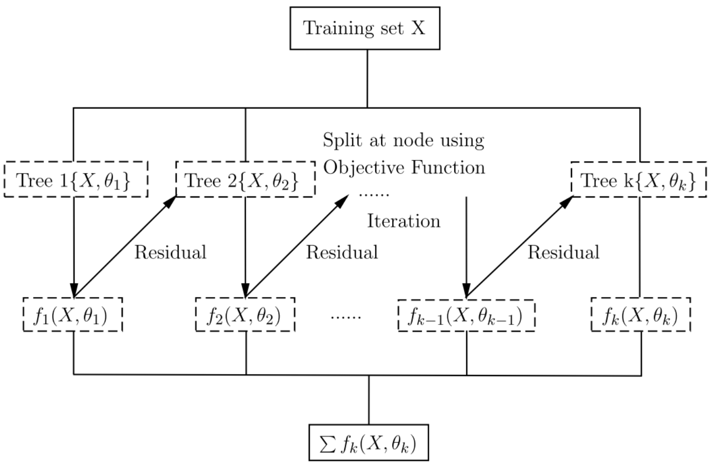
Support vector machines uses a function called kernel to calculate the inner product between a new observation and a training observation (James et al., 2013). The linear kernel function is classified as:

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij}x_{i'j},$$

We made two Support vector machines model, one using linear kernel with type = C-classification and other using random search with SVMradialSigma. The model with best accuracy was the one with linear kernel. We also made SVM models with random search and grid search as used by Lameski et al. (2015)

4. XGBoost:

XGboost is also known as extreme gradient boosted trees algorithm (Zhang et al., 2018). Chen and Guestrin (2016) proposed XGboost algorithm in 2016 and since then it have become one of the famous methods of machine learning (Zhang et al., 2018). It uses Gradient Boosted as the original model to combine weak learners to strong learners using multiple iterations. Residual from the previous predictor is used at each iteration to optimise the specified loss function (Zhang et al., 2018). For more details regarding XGboost refer to Zhang et al. (2018, p. 21025). Below figure shows the working of XGboost



Source: Zhang et al., 2018

We used grid search tuning method with 10-fold cross validation to apply the XGboost to the dataset. Grid Search has also been used by Sun (2020). The details of tuning grid and control function for the XGboost model are shown below:

```r
#xg boost tuning grid

xgboosttune <- expand.grid(nrounds=c(500,1000,1500),
                                      eta = c(0.01,0.05),
                                      max_depth = c(2,4,6),
                                      colsample_bytree = c(0.5,1),
                                      subsample = c(0.5,1),
                                      gamma = c(0,50),
                                      min_child_weight = c(0,20))

control_xgb <- trainControl(method = "cv",
                             number=10, #folds of cross validation
                             verboseIter = TRUE,
                            allowParallel = TRUE)



xgb <- train(absent_type ~ .,
             data = train,
             method = "xgbTree",
             trControl = control_xgb,
             tuneGrid = xgboosttune,
             verbose = TRUE)
```

5. Decision Tree:

Decision Tree performs classification without requiring domain knowledge or parameter settings (Wahid et al., 2019). We used the 10-fold cross validation method for the decision trees with grid search as tuning parameter.

Performance Measures:

Accuracy and Kappa values were used as performance measures for all the models. Accuracy is measured as:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

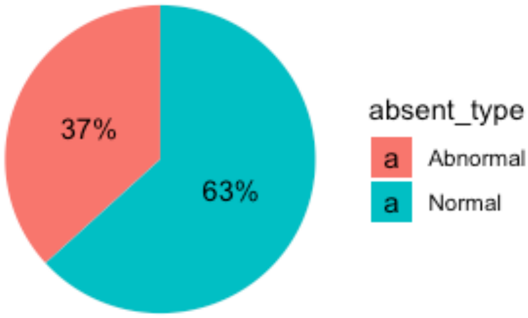
Source: Wahid et al., 209

Where TP denotes True Positive, TN denotes True Negative, FP denotes False Positive and FN denotes False Negative. Accuracy has been used as performance measure by various researchers (Wahid et al., 2019, Nath et al., 2022).

Kappa value also known as Cohen’s Kappa calculates the ratio between chance corrected agreement of accuracy in numerator and chance corrected perfect agreement in the denominator (Czodrowski, 2014). It provides an estimate of how better the agreement is over chance agreement (Czodrowski, 2014). The value ranges between -1 and +1 and the value below 0 indicates random guessing. Values greater than 0.3 are generally good fit (Czodrowski, 2014).

3. Findings:

It was discovered that the data set had 37% of absentees classified as Abnormal and 63% as normal.



Model Performance:

The model performance for each algorithm is shown below:

Algorithm	Accuracy	Kappa
LASSO Logistic Regression	0.8231	0.6224
XG Boost	0.8231	0.6195
Decision Tree	0.8231	0.6391
Random Forest	0.8095	0.5934
SVM	0.8095	0.5995

It was discovered that LASSO Logistic Regression, XG Boost and Decision Tree model performed better than Random Forest and SVM with Accuracy of 0.8231 for LASSO Logistic Regression, XGBoost and Decision Tree and 0.8095 for Random Forest and SVM. The kappa value is highest for the Decision tree model at 0.6391.

Variable Importance:

Lasso Logistic Regression:

The variable importance for LASSO Logistic Regression is shown below:

```
> varImp(model1)
glmnet variable importance

  only 20 most important variables shown (out of 70)

                     Overall
reason_for_absence27  100.00
reason_for_absence28   96.95
reason_for_absence23   94.54
disciplinary_fail1     91.15
reason_for_absence19   43.50
reason_for_absence22   32.31
reason_for_absence25   32.16
reason_for_absence18   23.26
reason_for_absence1    23.05
transport_expense      22.70
reason_for_absence10   21.68
reason_for_absence26   18.26
reason_for_absence13   17.60
social_drinker1        17.15
day_of_week5           16.64
reason_for_absence16   15.75
Education2             14.12
Son1                   14.04
reason_for_absence9    13.40
reason_for_absence6    12.74
```

Reason for Absence 27 which is a code for physiotherapy stands as the most important predictor for abnormal absenteeism. Followed by 28 which stands for dental consultation. Reason for absence 23 which is a code for medical consultation is the 3rd most important predictor. Employees with disciplinary failure of yes (1) accounts for 91.15 importance in the model. Further important predictors include reason for absence include 19 (Injury, poisoning and certain other consequences of external causes), 22 (patient follow up), 25 (laboratory examination), 18 (Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified). Apart from sickness specified as reason for absence. The transport expense, social drinker and Friday (day week 5) are considered to be the important predictors of abnormal absence.

Random Forest Variable Importance:

Important predictors of Random Forest best model are shown below:

```
> varImp(rf_grid)
rf variable importance

  only 20 most important variables shown (out of 68)

                       Overall
reason_for_absence28    100.00
reason_for_absence23     94.67
disciplinary_fail1       78.32
reason_for_absence27     64.16
avg_workload_day         55.57
transport_expense        48.00
ID                       46.90
hit_target               38.51
reason_for_absence19     33.26
reason_for_absence25     32.15
reason_for_absence22     28.46
service_time             24.68
Age                      23.21
Height                   22.46
BMI                      21.27
reason_for_absence13     21.00
Weight                   18.06
distance_residence_work  17.89
Son                      15.99
day_of_week4             15.96
```

In the random forest model, the reason for absence 28 (dental consultation) and 23 (medical consultation) are considered as the relatively important predictors which is different than the LASSO logistic regression model noted above. The disciplinary failure also has significant importance in predicting abnormal absences followed by reason for absence 27 (physiotherapy). Average workload in a day is found to have variable importance of 55.57 in predicting abnormal absences, followed by transport expense.

XG boost Variable Importance:

The variable importance of the XGboost best model is shown below:

```
> varImp(xgb)
xgbTree variable importance

  only 20 most important variables shown (out of 68)

                        Overall
reason_for_absence28     100.00
reason_for_absence23      98.03
disciplinary_fail1        80.63
reason_for_absence27      71.19
ID                        70.81
avg_workload_day          67.47
transport_expense         56.53
reason_for_absence19      46.82
Age                       40.08
hit_target                37.51
BMI                       33.17
service_time              32.78
Height                    31.44
reason_for_absence25      28.90
distance_residence_work   28.65
Weight                    28.52
reason_for_absence22      25.01
reason_for_absence13      24.46
Son                       21.07
reason_for_absence10      18.95
```

Reason for absence 28 and 23 are noted as the most important predictor of abnormal absences, along with disciplinary failure and average workload/day. Transport expense has variable importance of 56.53 in XGboost model.

Decision Tree Variable Importance:

The results of variable importance of decision tree model are provided below:

```
> varImp(modelDT)
rpart variable importance

  only 20 most important variables shown (out of 70)

                        Overall
reason_for_absence27    100.000
disciplinary_fail1       96.370
reason_for_absence28     72.660
reason_for_absence19     58.386
reason_for_absence22     51.308
transport_expense        44.952
reason_for_absence25     40.017
reason_for_absence23     32.503
BMI                      15.511
service_time             13.354
social_drinker1          12.014
distance_residence_work   8.829
reason_for_absence5       0.000
reason_for_absence6       0.000
month_of_absence11        0.000
reason_for_absence14      0.000
Pet1                      0.000
reason_for_absence17      0.000
day_of_week4              0.000
Pet8                      0.000
```

In the decision tree model, the reason for absence 27 (physiotherapy) has the highest importance followed by disciplinary failure. Apart from reason for absence 28, 19 and 22. Transport Expense is found to have variable importance of 44.952. BMI, Service time and social drinker attributes also contributed towards abnormal absence prediction.

4. Conclusion:

This research analysed the probability of employees committing abnormal absents on Brazilian Courier Company data set. The best predictor of abnormal absences is found to be decision tree model. This result differs with previous researchers who found performance of other algorithms better than decision trees. This could be due to the difference of tuning parameters employed. Wahid et al. (2019) found the accuracy of 82% using Gradient Boosted, 80.4% for Random forest and 79% with Tree Ensemble and decision trees. We found accuracy of 82% using LASSO logistic regression, XG boost and decision tree, and 80.9% for Random Forest. This shows that using grid search and 10-fold cross validation has yielded better predictive performance for most models.

Nath et al. (2022) who analysed the same data set got predictive Accuracy of 93.2% on MLR, 88.7% on SVM and 86.9% on random forest. This shows that it is possible to achieve the predictive accuracy score of more than what we reported on the same data set and our models have further room for improvement.

Wahid et al. (2019) analysed the same data set from a Brazilian Courier Company and applied Gradient Boosted Trees, Random Forest, Tree Ensemble and Decision Tree algorithms. They used 7 evaluation metrics such as Sensitivity, Specificity, Accuracy, True Positive, True Negative, False Positive and False Negative to measure the predictive performance of the algorithms. They reported predictive accuracy of 82% using Gradient Boosted Trees and 79% with Tree Ensemble.

This study noted medical reasons for absences, average workload/day, disciplinary failure, BMI, service time, transport expense, education level and distance from residence to work alongside other specified the results above as the most important predictors of abnormal absences. This information could be used by employees to control the abnormal absenteeism among their workforce. As discussed earlier, the abnormal absenteeism results in added cost and reduced benefit to an organization and controlling abnormal absenteeism is one of the main objectives of HRM. This study can be used by HRM professionals to understand the factors impacting abnormal absenteeism and take actions accordingly.

Although, this research has been done on a Brazilian data set, similar circumstances exist all across the globe (Ali Shah et al., 2020). The same models could also be trained on the local data set of any other geographical location in order to predict the abnormal absenteeism. This research adds to the growing literature on predicting employing absenteeism. The findings could be used by Human Resource Specialist and higher management of organizations in order to mitigate the factors impacting abnormal absenteeism. Kocakulah et al. (2016) suggested various measures that could be adapted in order to decrease abnormal absences. Keeping in view the important predictors proposed in this study, it is suggested that organizations could take following measures to address the issue (Kocakulah et al., 2016):

1. Creation of positive company culture
2. Increased work life balance and decreased workload for employees.
3. Medical assistance and employee assistantship programs.
4. Childcare and flexible scheduling

Appendix 1: Data set description

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
VIII Diseases of the ear and mastoid process
IX Diseases of the circulatory system
X Diseases of the respiratory system
XI Diseases of the digestive system
XII Diseases of the skin and subcutaneous tissue
XIII Diseases of the musculoskeletal system and connective tissue
XIV Diseases of the genitourinary system
XV Pregnancy, childbirth and the puerperium
XVI Certain conditions originating in the perinatal period
XVII Congenital malformations, deformations and chromosomal abnormalities
XVIII Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
XIX Injury, poisoning and certain other consequences of external causes
XX External causes of morbidity and mortality
XXI 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
6. Transportation expense
7. Distance from Residence to Work (kilometers)
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)
22. absent_type (new categorical variable created by classifying absenteeism time lower than 6 hour as “Normal” and above 6 hours as “Abnormal”

Appendix 2: Confusion Matrix of LASSO Logistic Regression Model

```
> confusionMatrix(predictionlasso, test$absent_type)
Confusion Matrix and Statistics

          Reference
Prediction Abnormal Normal
  Abnormal       42     14
  Normal         12     79

               Accuracy : 0.8231
                 95% CI : (0.7517, 0.8811)
    No Information Rate : 0.6327
    P-Value [Acc > NIR] : 3.54e-07

                  Kappa : 0.6224

 Mcnemar's Test P-Value : 0.8445

            Sensitivity : 0.7778
            Specificity : 0.8495
         Pos Pred Value : 0.7500
         Neg Pred Value : 0.8681
             Prevalence : 0.3673
         Detection Rate : 0.2857
   Detection Prevalence : 0.3810
      Balanced Accuracy : 0.8136

       'Positive' Class : Abnormal

> 
```

Appendix 3: Confusion Matrix of Random Forest Model

```
Confusion Matrix and Statistics

          Reference
Prediction Abnormal Normal
  Abnormal       41     15
  Normal         13     78

               Accuracy : 0.8095
                 95% CI : (0.7366, 0.8695)
    No Information Rate : 0.6327
    P-Value [Acc > NIR] : 2.417e-06

                  Kappa : 0.5934

 Mcnemar's Test P-Value : 0.8501

            Sensitivity : 0.7593
            Specificity : 0.8387
         Pos Pred Value : 0.7321
         Neg Pred Value : 0.8571
             Prevalence : 0.3673
         Detection Rate : 0.2789
   Detection Prevalence : 0.3810
      Balanced Accuracy : 0.7990

       'Positive' Class : Abnormal

> 
```

Appendix 4: SVM confusion Matrix

```
> confusionMatrix(svmpred, test$absent_type) #best mode
Confusion Matrix and Statistics

          Reference
Prediction Abnormal Normal
  Abnormal       43     17
  Normal         11     76

               Accuracy : 0.8095
                 95% CI : (0.7366, 0.8695)
    No Information Rate : 0.6327
    P-Value [Acc > NIR] : 2.417e-06

                  Kappa : 0.5995

 Mcnemar's Test P-Value : 0.3447

            Sensitivity : 0.7963
            Specificity : 0.8172
         Pos Pred Value : 0.7167
         Neg Pred Value : 0.8736
             Prevalence : 0.3673
         Detection Rate : 0.2925
   Detection Prevalence : 0.4082
      Balanced Accuracy : 0.8068

       'Positive' Class : Abnormal
```

Appendix 5: XGBoost Confusion Matrix

```
> confusionMatrix(predictxgb, test$absent_type) #best model
Confusion Matrix and Statistics

          Reference
Prediction Abnormal Normal
  Abnormal       41     13
  Normal         13     80

               Accuracy : 0.8231
                 95% CI : (0.7517, 0.8811)
    No Information Rate : 0.6327
    P-Value [Acc > NIR] : 3.54e-07

                  Kappa : 0.6195

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.7593
            Specificity : 0.8602
         Pos Pred Value : 0.7593
         Neg Pred Value : 0.8602
             Prevalence : 0.3673
         Detection Rate : 0.2789
   Detection Prevalence : 0.3673
      Balanced Accuracy : 0.8097

       'Positive' Class : Abnormal
```

Appendix 6: R code for the model

```
#get working directory
getwd()


#load the libraries

library(corrplot)
library(ggplot2)
library(dplyr)
library(readr)
library(readxl)
library(leaps)
library(ISLR2)
library(glmnet)
install.packages('el1071')
library(e1071)
library(caret)
#read the data

data <- read_excel("Absenteeism_at_work.xls")


#summary
summary(data)

#rename the colummns
colnames(data)[colnames(data) == "Reason for absence"] <- "reason_for_absence"
colnames(data)[colnames(data) == "Month of absence"] <- "month_of_absence"
colnames(data)[colnames(data) == "Day of the week"] <- "day_of_week"
colnames(data)[colnames(data) == "Transportation expense"] <- "transport_expense"
colnames(data)[colnames(data) == "Distance from Residence to Work"] <-
"distance_residence_work"
colnames(data)[colnames(data) == "Service time"] <- "service_time"
colnames(data)[colnames(data) == "Work load Average/day"] <- "avg_workload_day"
colnames(data)[colnames(data) == "Hit target"] <- "hit_target"
colnames(data)[colnames(data) == "Disciplinary failure"] <- "disciplinary_fail"
colnames(data)[colnames(data) == "Social drinker"] <- "social_drinker"
colnames(data)[colnames(data) == "Social smoker"] <- "social_smoker"
colnames(data)[colnames(data) == "Body mass index"] <- "BMI"
colnames(data)[colnames(data) == "Absenteeism time in hours"] <- "Absenteeism_time"

summary(data)

#convert factor variables to type factor

data$reason_for_absence <- as.factor(data$reason_for_absence)
data$month_of_absence <- as.factor(data$month_of_absence)
```

```
data$day_of_week <- as.factor(data$day_of_week)
data$Seasons <- as.factor(data$Seasons)
data$disciplinary_fail <- as.factor(data$disciplinary_fail)
data$Education <- as.factor(data$Education)
data$social_drinker <- as.factor(data$social_drinker)
data$social_smoker <- as.factor(data$social_smoker)
data$Pet <- as.factor(data$Pet) #coding number of  pets as a factor as well
data$Son <- as.factor(data$Son)

#outcome variable has absenteeism time in hour with mean of 6.924. We are gonnna divide
the absenteeism between normal/abnomral keeping 6 hour as the limit for normal

data$absent_type <- factor(ifelse(data$Absenteeism_time > 6, "Abnormal", "Normal"))


summary(data)


#look for missing values
#no missing values found in the data
sum(colnames(is.na))

missingValueCheck <- function(data)
{
  for (i in colnames(data))
  {
    print(i)
    print(sum(is.na(data[i])))
  }
  print("Total")
  print(sum(is.na(data)))
}
missingValueCheck(data)

###### EXPLORATORY ANALYSIS #####


#percentage of normal and abnormal absents
#37% of total absents are abnormal
absentplot <- data %>% group_by(absent_type) %>%
  count() %>% ungroup() %>% mutate(perc = `n`/sum(`n`)) %>%
  arrange(perc) %>%
  mutate(labels = scales::percent(perc))


summary(absentplot)
table(absentplot)
```

```
ggplot(data = absentplot, aes(x = "", y = perc, fill = absent_type)) +
  geom_col() +
  geom_text(aes(label = labels), position = position_stack(vjust = 0.5), show.legend = TRUE) +
  coord_polar(theta = "y") +
  theme_void()



#outliers analysis to remove any unusual values

#for factors
boxplot(data$reason_for_absence) #no outlier noted
boxplot(data$month_of_absence) #no outlier noted
boxplot(data$day_of_week) # no outlier noted
boxplot(data$Seasons) #no outlier noted
boxplot(data$service_time)


boxplot(data$Education)
table(data$Education) # most of the people are educated till high school but we will keep
these in the data

boxplot(data$disciplinary_fail)
boxplot(data$Pet)
table(data$Pet)
#for numeric variables
hist(data$transport_expense)
hist(data$distance_residence_work)
hist(data$service_time, labels = TRUE) #few outliers detected
hist(data$Age, labels = TRUE)
hist(data$avg_workload_day, labels = TRUE)
hist(data$hit_target, labels = TRUE)
hist(data$Son)





#visualisations




#dropping the absenteeism time column

summary(modeldata)

modeldata <- subset(data, select = -c(Absenteeism_time))
```

```
#### MODEL Building

set.seed(1992)

index <- createDataPartition(modeldata$absent_type, p = 0.8, list = FALSE, times=1)

train <- modeldata[index,] #index reference by rows

test <- modeldata[-index,]




#### Specify and train Lasso Regression model

ctrlspecLASSO <- trainControl(method="cv", number = 10,
                       savePredictions = "all",
                       summaryFunction = twoClassSummary,
                       classProbs = TRUE)


#create a vector for lambda values

lambda_vector <- 10^seq(5, -5, length=500)

#setseed
set.seed(1992)

#Specify the Lasso  regression model using training data and 10 fold cross validation



model1 <- train(absent_type ~ .,
           data = train,
           preProcess=c("center", "scale"),#preprocess the predictor variable to scale them
           method="glmnet",
           metric = "ROC",
           tuneGrid=expand.grid(alpha=1, lambda = lambda_vector),
           trControl=ctrlspecLASSO, na.action=na.omit,
           family = "binomial") #train control to specify our 10 fold cross validation function

model1$bestTune$lambda

round(coef(model1$finalModel, model1$bestTune$lambda), 3)
```

```
print(model1)


plot(log(model1$results$lambda),
     model1$results$ROC,
     xlab = "log lambda",
     ylab = "ROC",
     xlim = c(1,20),
     ylim = c(0.0, 1))

#variable importance
varImp(model1)
plot(varImp(model1))

install.packages("vip")
library(vip)

#exporting variable importance as a table
lassovarimp <- as.data.frame(vi(model1))

#keeping importance greater than 0

lassovarimp1 <- lassovarimp %>% filter(Importance > 0)


ggplot(varImp(model1))

ggplot(data = lassovarimp1, mappoing=aes(x=Importance, y=variable), stat = summary) +
  geom_bar()


####predicting the performance
predictionlasso <- predict(model1, newdata=test)

plot(x=predictionlasso, y=test$absent_type)
abline(a=0, b=1)


# Model Performance/Accuracy

confusionMatrix(predictionlasso, test$absent_type)
```

```
######RANDOM FOREST###################
set.seed(1992)
ctrlspecRF <- trainControl(method="cv", number = 10,
                    search = "random",
                    savePredictions = T)

#applying a random forest model

randomforest <- train(absent_type ~ .,
                data = train,
                method = "rf",
                trControl = ctrlspecRF, tuneLength = 10,
                ntree=1000)

print(randomforest)

randomforest$bestTune

plot(randomforest)
plot(varImp(randomforest, scale = F), main = "Variable importance for RF")

#model prediction

rfprediction <- predict(randomforest, newdata = test)

confusionMatrix(rfprediction, test$absent_type)


#applying a random forest model with more number of trees


randomforest2 <- train(absent_type ~ .,
                data = train,
                method = "rf",
                trControl = ctrlspecRF, tuneLength = 17,
                ntree=5000)




randomforest2$bestTune


plot(varImp(randomforest, scale = F), main = "Variable importance for RF")
```

```
#model prediction

rfprediction2 <- predict(randomforest2, newdata = test)

confusionMatrix(rfprediction2, test$absent_type)


#random forest model with grid search after specifying tuning grid with repeated 10 fold CV

controlRFgridsearch <- trainControl(method = 'repeatedcv',
                         number = 10,
                         repeats = 3,
                         search = 'grid'
                          )

tunegrid_rf <- expand.grid(.mtry = (1:30))

rf_grid <- train(absent_type ~ .,
            data = train,
            method = "rf",
            tuneGrid = tunegrid_rf,
            trControl = controlRFgridsearch)

print(rf_grid)

rf_grid$bestTune

plot(rf_grid)

varImp(rf_grid)

 #model prediction

rfgridpred<- predict(rf_grid, newdata = test)

confusionMatrix(rfgridpred, test$absent_type) #best mode




################SUPPORT VECTOR MACHINES#################

svm <- svm(formula = absent_type ~ .,
        data = train,
        type = 'C-classification',
```

```
        kernel = 'linear')
print(svm)


svmpred <- predict(svm, newdata = test)

confusionMatrix(svmpred, test$absent_type) #best mode



#svm using random search

set.seed(1992)

#specify the control function for random search
controlsvmrandom <- trainControl(method = 'cv',
                        number = 10,
                        search = 'random',
                       savePredictions = TRUE)


#specify SVM model

svmrandom <- train(absent_type ~ .,
             data = train,
             method = "svmRadialSigma",
             trControl = controlsvmrandom,
             tuneLength = 20)

svmrandom$bestTune #values of signma for the best model as per random search is
0.0000008190721 and c is 0.04


svmpredrandom <- predict(svmrandom, newdata = test)

confusionMatrix(svmpredrandom, test$absent_type)



#SVM using grid search

controlsvmgrid <- trainControl(method = 'cv',
                      number = 10,
                      savePredictions = TRUE)

tuneGridsvm=expand.grid(
  .sigma=seq(0.0000000000491661, 0.0000000000120000, length = 20),
```

```
 .C=seq(0.01, 0.06, length = 20))


svmgrid <- train(absent_type ~ .,
                 data = train,
                 method = "svmRadialSigma",
                 trControl = controlsvmgrid,
                 tuneGrid = tuneGridsvm)

svmpred <- predict(svmgrid, newdata = test)

confusionMatrix(svmpred, test$absent_type) #best mode

varImp(svmgrid)



######gradient boosted tree#


#xg boost tuning grid

xgboosttune <- expand.grid(nrounds=c(500,1000,1500),
                           eta = c(0.01,0.05),
                           max_depth = c(2,4,6),
                           colsample_bytree = c(0.5,1),
                           subsample = c(0.5,1),
                           gamma = c(0,50),
                           min_child_weight = c(0,20))

control_xgb <- trainControl(method = "cv",
                            number=10, #folds of cross validation
                            verboseIter = TRUE,
                           allowParallel = TRUE)




xgb <- train(absent_type ~ .,
             data = train,
             method = "xgbTree",
             trControl = control_xgb,
             tuneGrid = xgboosttune,
             verbose = TRUE)

predictxgb <- predict(xgb, newdata = test)
confusionMatrix(predictxgb, test$absent_type) #best model
```

```
varImp(xgb)
plot(predictxgb)



#### DECISION TREE MODEL###

install.packages("rattle")
library(rattle)
library(caret)

ctrlDT <- trainControl(method = "cv", #cross validation
                 number = 10)   #10-fold cross validation

grid_DT <- data.frame(cp = seq(0.02, .2, .02))

modelDT <- train(absent_type~., data = train, method = 'rpart',
            trControl = ctrlDT,
            tuneGrid = grid_DT)


modelDT

#plot dt
fancyRpartPlot(modelDT$finalModel, sub = NULL)


#predictive performance
predictDT <- predict(modelDT, newdata = test)
confusionMatrix(predictDT, test$absent_type)

varImp(modelDT)
```

References:

1. Ajmi, S.Q., 2020. Predicting Absenteeism at Work Using Machine Learning Algorithms. MJPS, 7(1).
2. Ali Shah, S.A., Uddin, I., Aziz, F., Ahmad, S., Al-Khasawneh, M.A. and Sharaf, M., 2020. An enhanced deep neural network for predicting workplace absenteeism. Complexity, 2020.

3. Bycio, P., 1992. Job performance and absenteeism: A review and meta-analysis. Human relations, 45(2), pp.193-220.
4. Chen, T. and Guestrin, C., 2016, August. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).
5. Cohen, A. and Golan, R., 2007. Predicting absenteeism and turnover intentions by past absenteeism and work attitudes: An empirical examination of female employees in long term nursing care facilities. Career Development International.
6. Cucchiella, F., Gastaldi, M. and Ranieri, L., 2014. Managing absenteeism in the workplace: the case of an Italian multiutility company. Procedia-Social and Behavioral Sciences, 150, pp.1157-1166.
7. Czodrowski, P., 2014. Count on kappa. Journal of computer-aided molecular design, 28(11), pp.1049-1055.
8. Dogruyol, K. and Sekeroglu, B., 2019, August. Absenteeism prediction: a comparative study using machine learning models. In International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions (pp. 728-734). Springer, Cham.
9. Halbesleben, J.R., Whitman, M.V. and Crawford, W.S., 2014. A dialectical theory of the decision to go to work: Bringing together absenteeism and presenteeism. Human Resource Management Review, 24(2), pp.177-192.
10. Haswell, M., 2003. Dealing with employee absenteeism. Management Services, 47(12), pp.16-17.
11. James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
12. Kaleta, S. and Edward, A., 2003. Here today, gone tomorrow: Employee absence and the evaporating workforce. National Underwriter, 1.
13. Kocakulah, M.C., Kelley, A.G., Mitchell, K.M. and Ruggieri, M.P., 2016. Absenteeism problems and costs: causes, effects and cures. International Business & Economics Research Journal (IBER), 15(3), pp.89-96.
14. Lameski, P., Zdravevski, E., Mingov, R. and Kulakov, A., 2015. SVM parameter tuning with grid search and its impact on reduction of model over-fitting. In Rough sets, fuzzy sets, data mining, and granular computing (pp. 464-474). Springer, Cham.
15. Martiniano, A., Ferreira, R. P., Sassi, R. J., & Affonso, C. (2012). Application of a neuro fuzzy network in prediction of absenteeism at work. In Information Systems and Technologies (CISTI), 7th Iberian Conference on (pp. 1-4). IEEE.
16. Meier, L., Van De Geer, S. and Bühlmann, P., 2008. The group lasso for logistic regression. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70(1), pp.53-71.
17. Nanjundeswaraswamy, T., 2016. An empirical study on absenteeism in Garment industry. Management Science Letters, 6(4), pp.275-284.
18. Nath, G., Harfouche, A., Coursey, A., Saha, K.K., Prabhu, S. and Sengupta, S., 2022. Integration of a machine learning model into a decision support tool to predict absenteeism at work of prospective employees. arXiv preprint arXiv:2202.03577.
19. Oliveira, E.L.D., Torres, J.M., Moreira, R.S. and Lima, R.A.F.D., 2019, April. Absenteeism prediction in call center using machine learning algorithms. In World Conference on Information Systems and Technologies (pp. 958-968). Springer, Cham.

20. Ramadhan, M.M., Sitanggang, I.S., Nasution, F.R. and Ghifari, A., 2017. Parameter tuning in random forest based on grid search method for gender classification based on voice frequency. DEStech Transactions on Computer Science and Engineering, 10.
21. Sarica, A., Cerasa, A. and Quattrone, A., 2017. Random forest algorithm for the classification of neuroimaging data in Alzheimer's disease: a systematic review. Frontiers in aging neuroscience, 9, p.329.
22. Skorikov, M., Hussain, M.A., Khan, M.R., Akbar, M.K., Momen, S., Mohammed, N. and Nashin, T., 2020, December. Prediction of Absenteeism at Work using Data Mining Techniques. In 2020 5th International Conference on Information Technology Research (ICITR) (pp. 1-6). IEEE.
23. Sun, L., 2020, December. Application and improvement of xgboost algorithm based on multiple parameter optimization strategy. In 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE) (pp. 1822-1825). IEEE.
24. Truman, D., 2003. Ohio companies try to encourage employees not to take sick days. Knight Ridder Tribune Business News, 1.
25. UCI. No date. Absenteeism at work data set. Source: online. https://archive.ics.uci.edu/ml/datasets/Absenteeism+at+work. Accessed: 11/05/2022.
26. Wahid, Z., Satter, A.Z., Al Imran, A. and Bhuiyan, T., 2019, January. Predicting absenteeism at work using tree-based learners. In Proceedings of the 3rd International Conference on Machine Learning and Soft Computing (pp. 7-11).
27. Wang, H., Xu, Q. and Zhou, L., 2015. Large unbalanced credit scoring using lasso-logistic regression ensemble. PloS one, 10(2), p.e0117844.
28. Zhang, D., Qian, L., Mao, B., Huang, C., Huang, B. and Si, Y., 2018. A data-driven design for fault detection of wind turbines using random forests and XGboost. Ieee Access, 6, pp.21020-21031.