Employee Absenteeism Analysis

*Yatin Vij*

*19th November 2018*

**Contents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [**1**](#page3) | [**Introduction**](#page3) | | | **2** |
|  | [1.1](#page3) | [Problem Statement](#page3) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 2 |
|  | [1.2](#page3) | [Data](#page3) . | . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 2 |
| [**2**](#page5) | [**Methodology**](#page5) | | | **4** |
|  | [2.1](#page5) | [Exploratory Data Analysis](#page5). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 4 |
|  |  | [2.1.1](#page5) | [Missing Value Analysis](#page5) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 5 |
|  |  | [2.1.2](#page5) | [Outlier Analysis](#page5) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 5 |
|  |  | [2.1.3](#page11) | [Feature Selection](#page11) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 9 |
|  |  | [2.1.](#page11)4 | [Feature Engineering](#page11). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 10 |
|  | [2.2](#page13) | [Modeling](#page13) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 11 |
|  |  | [2.2.1](#page13) | [Model Selection](#page13) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 11 |
|  |  | [2.2.2](#page13) | [Multiple Linear Regression](#page13) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 11 |
|  |  | [2.2.3](#page15) | Time Series Modeling. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 12 |
| **3** | **Conclusion** | |  | **13** |
|  | [3.1](#page16) | [Model Evaluation](#page16) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . | | 13 |
|  | [3.2](#page17) | [Model Selection](#page17) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 14 |
| [**Appendix A - Extra**](#page24) **Figures** | | | | **15** |
| [**Appendix B - Python Code**](#page24) | | | | **15** |
| [**Appendix A - R Code**](#page24) | | | | **18** |
| [**References**](#page30) | | |  | **29** |

**Chapter 1**

**Introduction**

**1.1 Problem Statement**

The objective of this case study is to analyze the given Employee Absenteeism Data and answer the following questions using the insights:

* What changes company should bring to reduce the number of absenteeism?
* How much losses every month can we project in 2011 if same trend of absenteeism continues?

**1.2 Data**

The given data is a Time series Multivariate dataset with 21 attributes with missing values. Our task is to analyze the dataset and find insights to help answer the questions posed to us by the company.

Given below is a sample of the data set that we are using for building the prediction model:

Table 1.1: Sample Data (Columns: 1-7)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Reason for absence | Month of absence | Day of the week | Seasons | Transportation expense | Distance from Residence to Work |
| 11 | 26 | 7 | 3 | 1 | 289 | 36 |
| 36 | 0 | 7 | 3 | 1 | 118 | 13 |
| 3 | 23 | 7 | 4 | 1 | 179 | 51 |
| 7 | 7 | 7 | 5 | 1 | 279 | 5 |
| 11 | 23 | 7 | 5 | 1 | 289 | 36 |

Table 1.2: Sample Data (Columns: 8-14)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Service time | Age | Work load Average/day | Hit target | Disciplinary failure | Education | Son |
| 13 | 33 | 239,554 | 97 | 0 | 1 | 2 |
| 18 | 50 | 239,554 | 97 | 1 | 1 | 1 |
| 18 | 38 | 239,554 | 97 | 0 | 1 | 0 |
| 14 | 39 | 239,554 | 97 | 0 | 1 | 2 |
| 13 | 33 | 239,554 | 97 | 0 | 1 | 2 |

Table 1.3: Sample Data (Columns:15-21)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Social drinker | Social smoker | Pet | Weight | Height | Body mass index | Absenteeism time in hours |
| 1 | 0 | 1 | 90 | 172 | 30 | 4 |
| 1 | 0 | 0 | 98 | 178 | 31 | 0 |
| 1 | 0 | 0 | 89 | 170 | 31 | 2 |
| 1 | 1 | 0 | 68 | 168 | 24 | 4 |
| 1 | 0 | 1 | 90 | 172 | 30 | 2 |

As you can see in the table below we have the following 20 variables, using which we can use to analyze the absenteeism data:

Table 1.4: Predictor Variables

S.No. Predictor

* ID

2 Reason for absence

3 Month of absence

4 Day of the week

5 Seasons

6 Transportation expense

7 Distance from Residence to Work

8 Service time

9 Age

1. Work load Average/day
2. Hit target
3. Disciplinary failure
4. Education
5. Son
6. Social drinker
7. Social smoker
8. Pet
9. Weight
10. Height
11. Body mass index

The details of data attributes in the dataset are as follows:

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
  + medical consultation
  + blood donation
  + laboratory examination
  + unjustified absence
  + physiotherapy
  + dental consultation

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 (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)

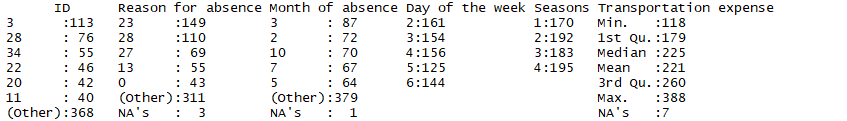
18. Weight

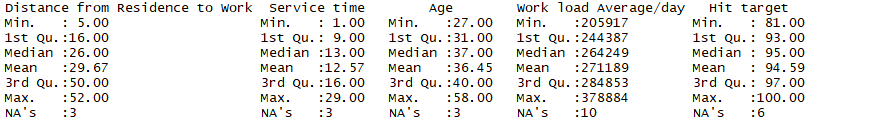
19. Height

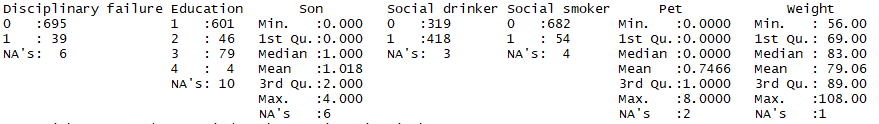
20. Body mass index

21. Absenteeism time in hours (target)

Let’s now look at the structure of the data given to us.







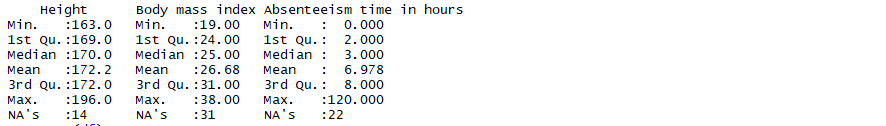


Fig 1.1: Structure of the Data

As we can see from the above figure, the data contains missing values in the following attributes:

* Reason for Absence
* Month of Absence
* Transportation Expense
* Distance from Residence to Work
* Service Time
* Age
* Work load Average/ day
* Hit Target
* Disciplinary Failure
* Education
* Son
* Social Drinker
* Social Smoker
* Pet
* Weight
* Height
* Body Mass Index
* Absenteeism time in hours

**Chapter 2**

**Methodology**

**2.1** **Exploratory Data Analysis:**

This phase is perhaps the most time consuming of all the phases as it deals with the following tasks:

* Exploring the data and all the variables that are present.
* Cleaning the data
  + Missing Value Analysis
  + Outlier Analysis
  + Feature Selection
  + Feature Scaling
* Visualizing the data

We have already explored the data we are dealing with in the above section where we listed all the variables we have and what they mean in the dataset. We saw that a few attributes have missing values present in them so let’s first impute these missing values before moving on.

**2.1.1 Missing Value Analysis**

In this section, we will be imputing the missing values present in the dataset. Upon looking at the data, we found that the data contains values for 36 individuals over a period of time, so for imputing the values more appropriately we can subset the data according to individuals and then use the imputation techniques for the subset.

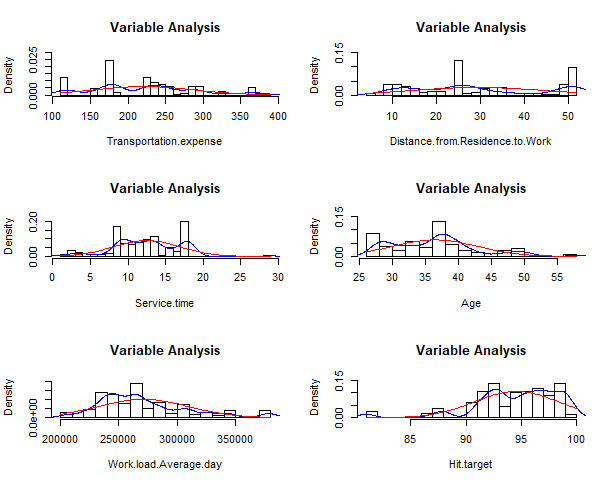
Let’s first impute the categorical attributes, for these variables we’ll be imputing the missing values with the mode of the attribute and if the mode is more than one, then we fill the missing values using the forward fill method.

* Reason for Absence
* Month of Absence
* Disciplinary Failure
* Education
* Social Drinker
* Social Smoker

For the numerical attributes, we’ll be using KNN to find the mean value of the attribute for the n neighbors and imputing the missing value with the obtained value.

* Transportation Expense
* Distance from Residence to Work
* Service Time
* Age
* Work load Average/ day
* Hit Target
* Son
* Pet
* Weight
* Height
* Body Mass Index
* Absenteeism time in hours

Now that we have imputed the missing values, let’s move forward with visualizing all the variables present in the data. As we have a mixture Numerical as well as Categorical data, we will be using different methods to visualize each. As we have to analyze the absenteeism of the employees, therefore the Target Variable will be Absenteeism time in hours.



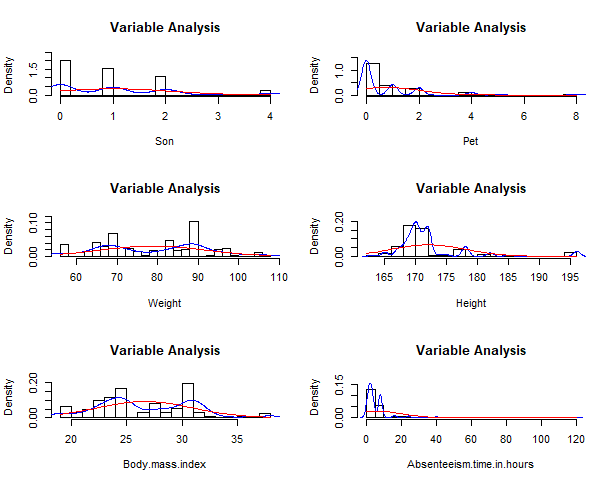
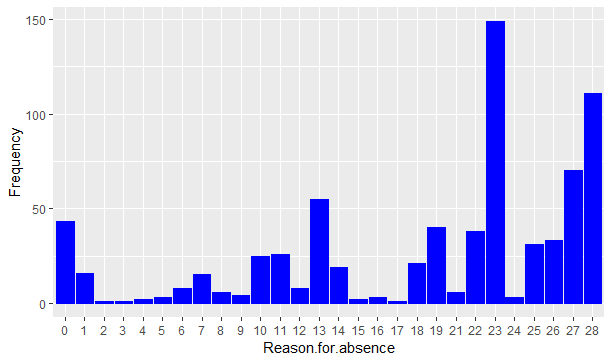
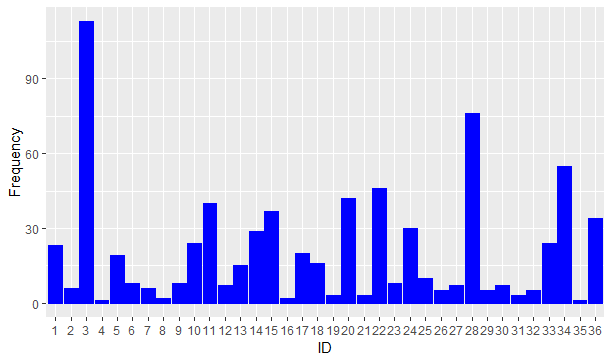
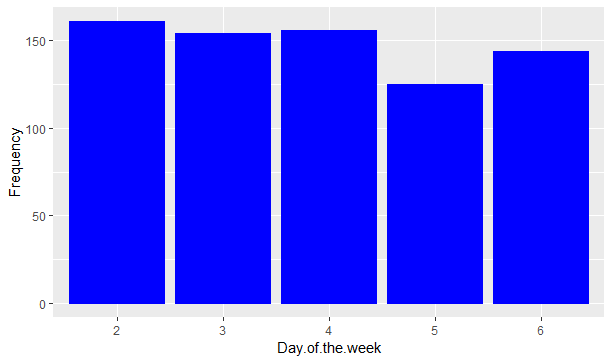
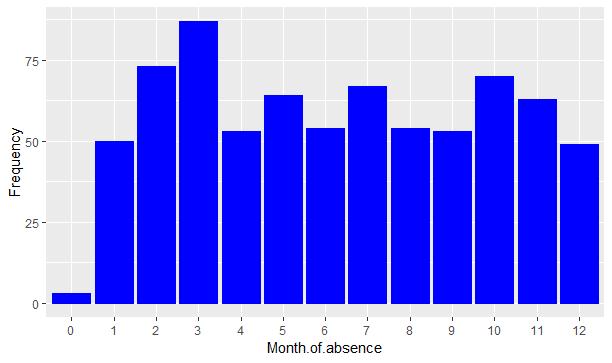


Fig 2.1 Probability Distribution Plot for all the numeric Variables

Fig 2.1 visualizes the probability distribution functions of the numerical variables i.e. Transportation Expense, Distance from Residence to Work, Service Time, Age, Work load Average/ day, Hit Target, Son, Pet, Weight, Height, Body Mass Index, Absenteeism time in hours.

Now let’s visualize the categorical variables using a bar plot of each variable w.r.t. frequency of each class of the variable.





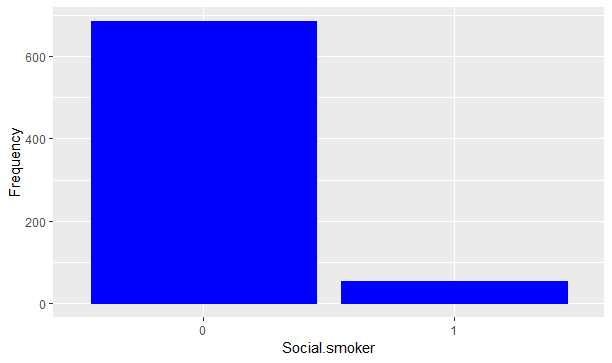
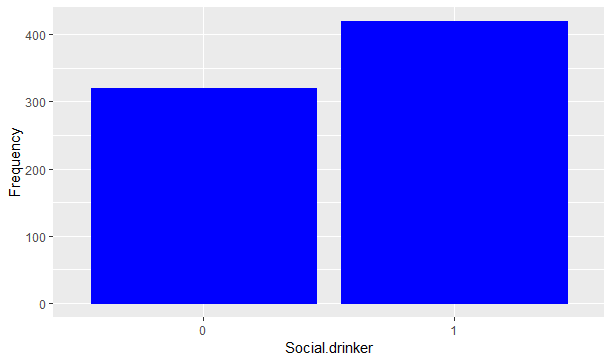
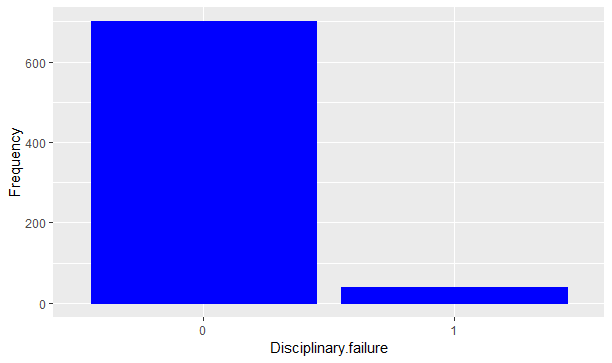
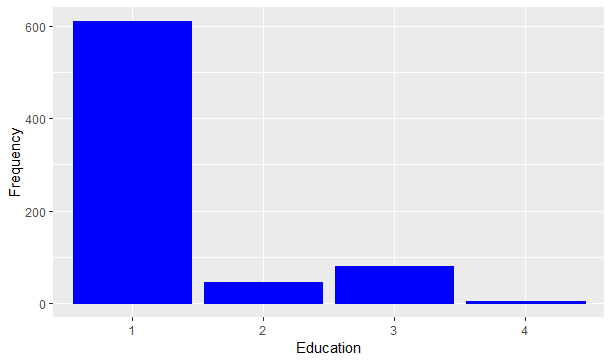
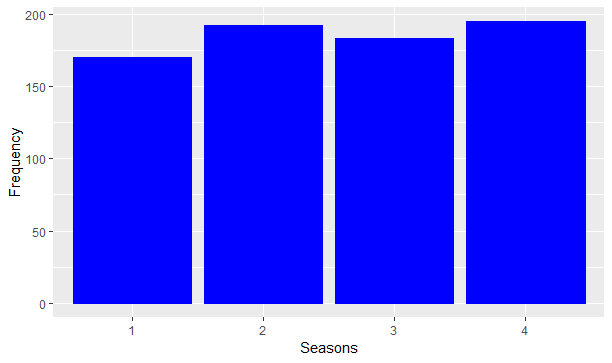
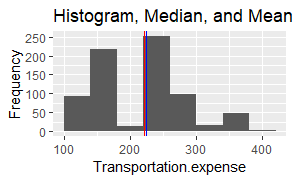
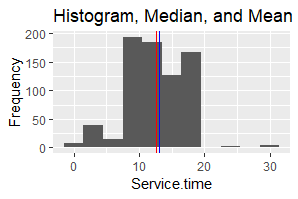
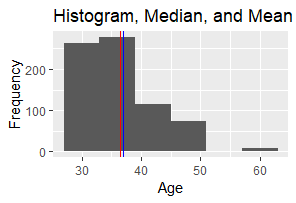
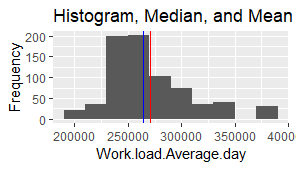
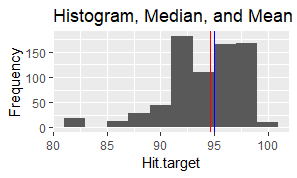
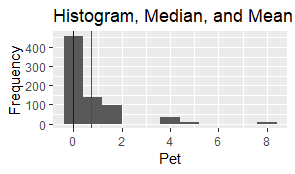


Fig 2.2 Bar Plot of Frequency Count for all classes of the Categorical Variables

**2.1.2 Outlier Analysis**

As we can see from fig 2.1, the variables Transportation Expense, Service Time, Age, Work load Average/ day, Hit Target, Pet, Height, Absenteeism time in hours are skewed. This is most likely due to the presence of Outliers in these variables. To visualize the effect of skew, let’s look at Fig 2.3 the Histogram, Median, and Mean plot of these variables.

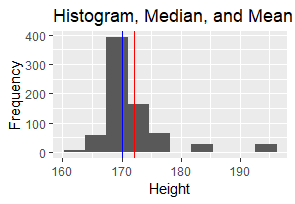
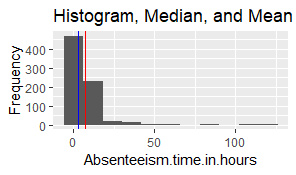
 

Fig 2.3 Histogram, Median and Mean of Numerical Variables

In Fig 2.3, the red line represents the mean and the blue line represents the median. As we can see, the mean is slightly displaced from the position of the median due to the presence of outliers in these variables.

As for the categorical variables, we can see that in the variables Reason for absence and Month of Absence there are values for class which does not exist i.e. Month has no 0 month and Reason for absence has no 0 class. This implies that these values are outliers and need to be imputed. To impute these, we simply replace all the 0 values with mode of the Month of absence attributes and for the Reason for absence as we can see from the Fig 2.2, it’s missing the value 20, this means the class 0 has been misclassified as 0 so we will be replacing 0 to 20.

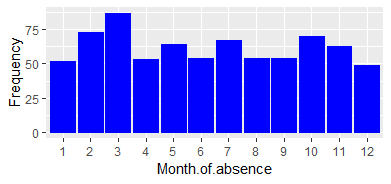
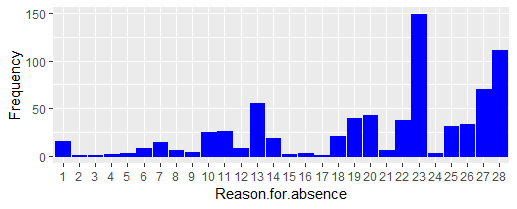
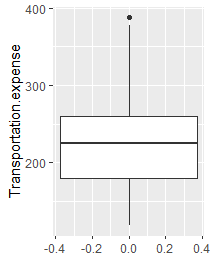
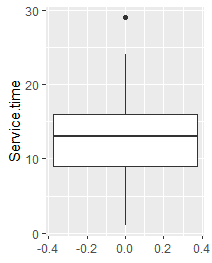
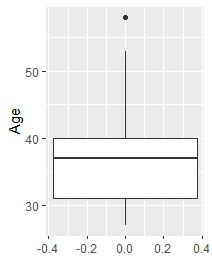
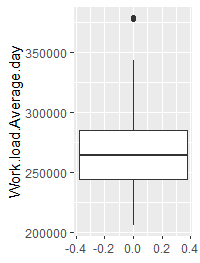
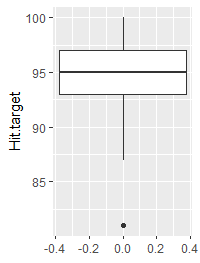
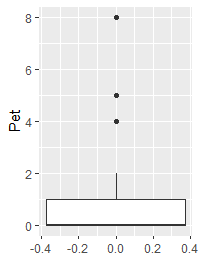


Fig 2.3.1 Frequency Bar Plot for Categorical Variables without outliers

Fig 2.3.1, shows the categorical variables after outlier imputation.

To visualize the outliers, let’s construct the boxplots of all these variables:

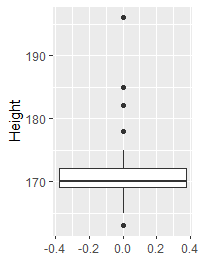
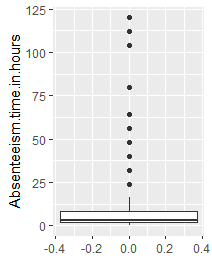
 

Fig 2.3.2 Boxplots for Numerical Variables

For the numerical variables (Fig 2.3.2), the boxplot is constructed for all the variables with only the variable itself on the y-axis.

As we can see from Fig 2.3.2, the outliers are present in Transportation Expense, Service Time, Age, Work load Average/ day, Hit Target, Pet, Height, and Absenteeism time in hours. After looking more carefully in these variables, we found that only Service Time & Absenteeism time in hours contain unrealistic values all other outliers are pretty possible so we will be leaving them as they are. Now we will be treating these outliers.

In Fig 2.4.1, we can see the outliers and the histograms of these variables before imputing the outliers.

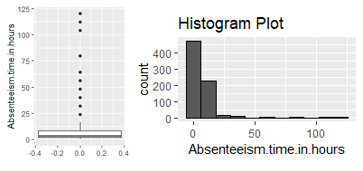
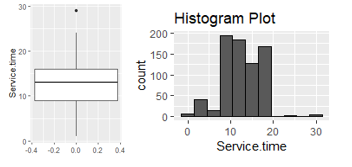


Fig 2.4.1 Boxplot & Histogram for Numerical Variables with outliers

After removing the outlier from the data, the histograms now look like shown in Fig 2.4.2. We get a much more normal distribution for the variables.

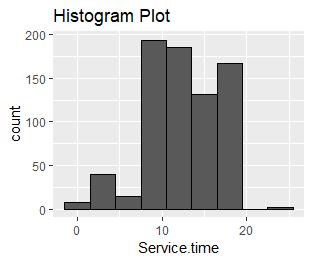
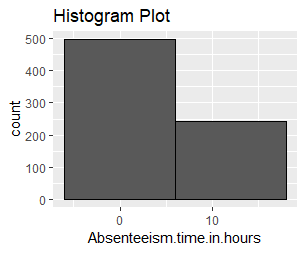
 

Fig 2.4.2 Histogram for Numerical Variables after removing outliers

Now we can move forward with the next step, Feature Selection.

**2.1.2 Feature Selection**

Now that we have cleaned the data, before stepping into modelling we will first identify the variables that are the most important for the analysis. We are dealing with two types variables i.e. Categorical and Numerical.

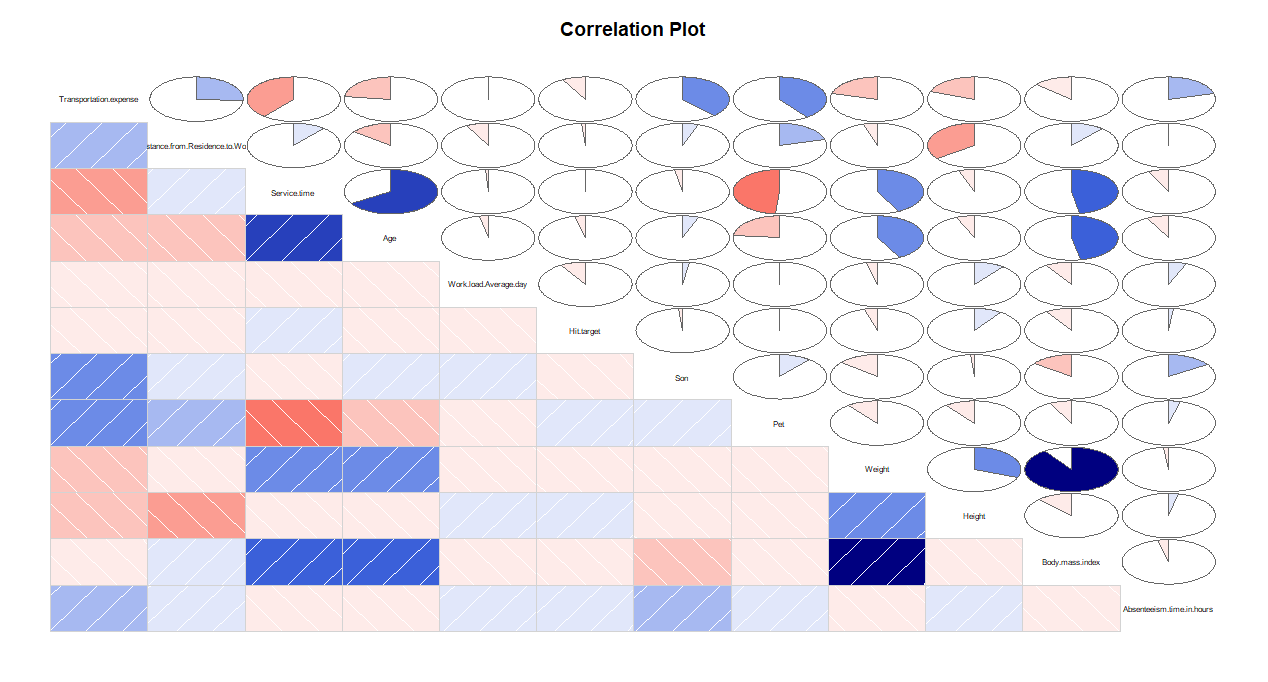


Fig 2.5 Correlation Plot for the Numerical Variables

For the numerical variables, we conduct the correlation analysis to find the highly correlated variables. As shown in Fig 2.5, the variables that are highly correlated are Weight – Body Mass Index and Service Time - Age. So we will be removing Body Mass Index and Age variables from our dataframe as they individually do not add any more information for the analysis when their correlated counterparts are present. Moreover, we will be removing the ID variable as well as it would not be contributing to the overall analysis.

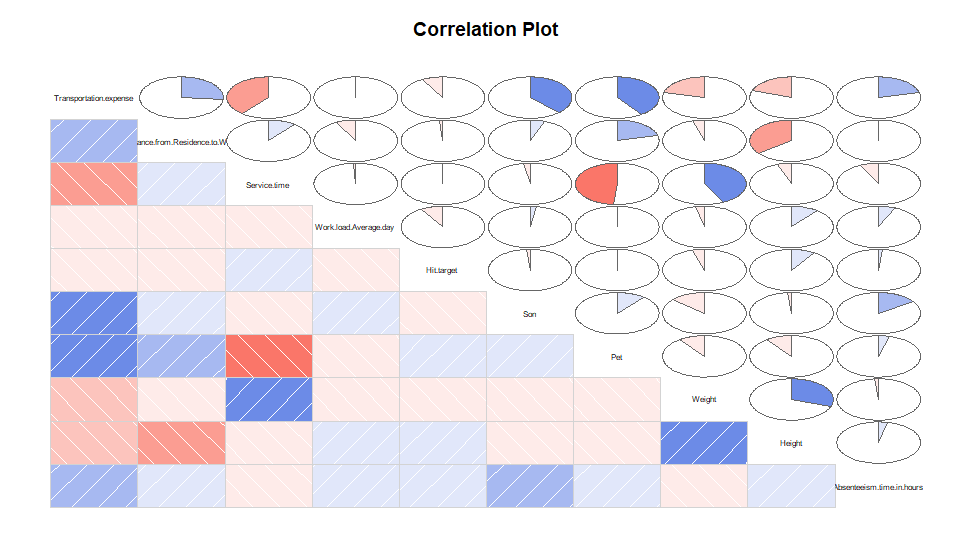


Fig 2.6 Correlation Plot for the Numerical Variables after removing Correlated variables

For the categorical variables, we will be using ANOVA to identify the significant categorical variables.















Fig 2.7 ANOVA of all the Variables w.r.t. Target

As we can see, the variables Day of the week, Seasons, and Social Smoker can be removed further from our dataframe. After this, we convert the categorical variables to numerical using One hot encoding technique.

For further reducing the variables, we use the Variance Inflation Factor (VIF) to find the variables with high VIF which would suggest the best variables that can be selected.

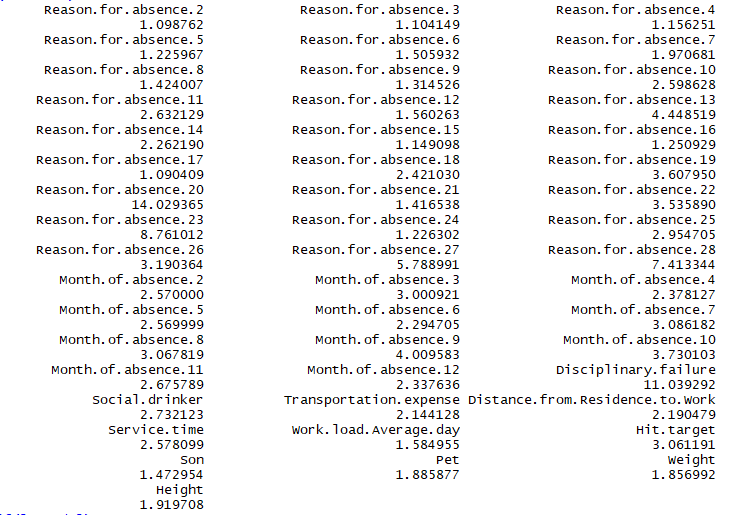


Fig 2.7 VIF of all the Variables

We take the threshold VIF value to be 5, so from the above figure we observe that Reason.for.absence.20 has the highest VIF and so we remove it and re-compute the VIF as shown in Fig 2.8.

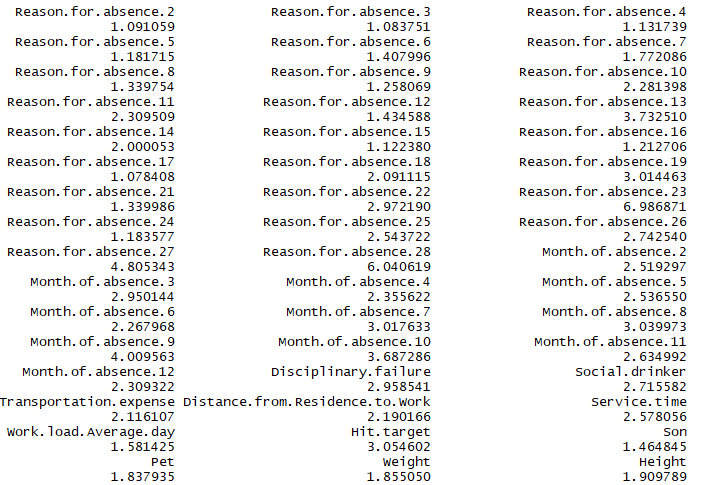


Fig 2.8 VIF of all the Variables after removal of the variable

Now we remove the Reason.for.absence.23 variable, and re-compute and find that all the VIFs are under the threshold value so we proceed forward. We find that all the variables have the VIF under the threshold and thus we can stop.

As VIF helps reduce variables for linear regression, we can similarly use Random Forest Model to find important variables using backward elimination method. The following figure shows the important variables that passed the threshold value of 6.

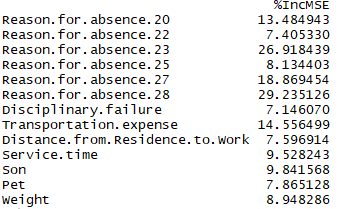


Fig 2.9 Random Forest Importance of the Variables after Backward Elimination

These variables will be used when we’ll be constructing the Random Forest model.

So all the variables that have no effect on the analysis or that don’t add any more information to the model and thus can be removed from the dataframe are:

* Body Mass Index
* Season
* Day of the week
* Social Smoker
* Age

**2.1.3 Feature Engineering**

To further help our analysis, we create a new feature named as Utilization. This new feature is calculated as the difference in Service time and Absenteeism time whole divided by the Service Time. This gives us an idea of the time utilization and thus help in guessing the loss. We are now ready to step into Model Development.

**2.2 Modeling**

**2.2.1 Model Selection**

After completing the Exploratory Data Analysis and pre-processing the data, we now understand the data much better and can use this understanding to develop prediction models. In our dataset, the dependent variable, Absenteeism time in hours, is a numerical variable so we will be using the Regression analysis.

For Regression Analysis, we will be using the Multiple Linear Regression. As it is a time series multivariate data, we will be converting the data into time series based on each month and then use Vector Auto Regression (VAR) model to fit and forecast the trend for the next 12 months. Let’s start building model from the simplest to more complex. Therefore, we start with Multiple Linear Regression.

**2.2.2 Multiple Linear Regression**

The regression was performed on the full dataset without converting it into time series. The below multiple linear regression model was constructed using backward elimination technique at 5% significance level. The remaining variables are the only ones affecting the absenteeism time.

As you can see in Fig 2.7, the Adjusted R-squared value, we can explain only about 37% of the data using our multiple linear regression model. After 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.

Looking at the significance values of the predictors, we can see that each predictor is contributing at 5% significance level.

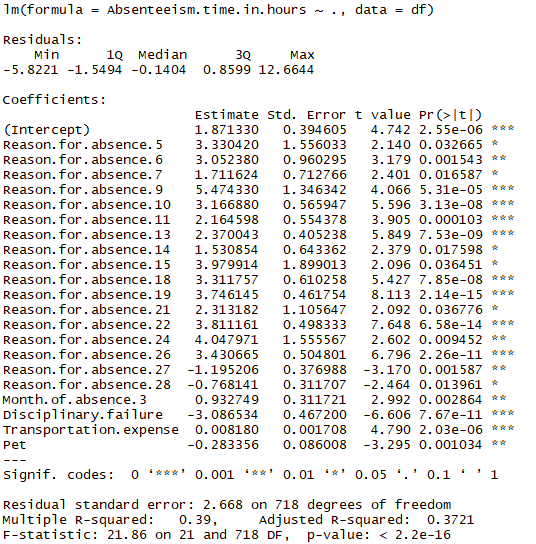


Fig 2.7 Linear Regression Model Summary

**2.2.3** **Time Series Modelling**

Let’s convert the dataframe into a timeseries and analyze the trends in the data to better answer the questions posed to us.

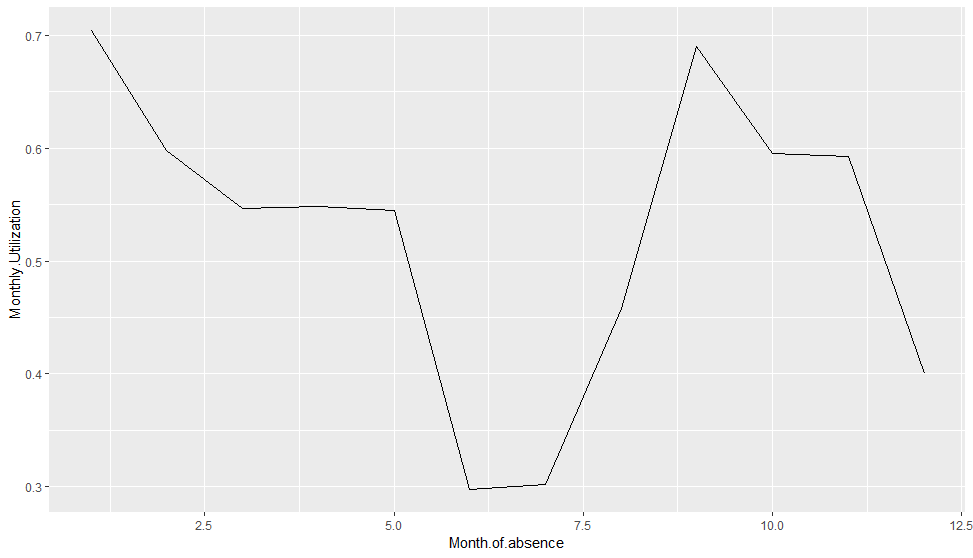


Fig 2.8 Line Plot of Monthly Utilization

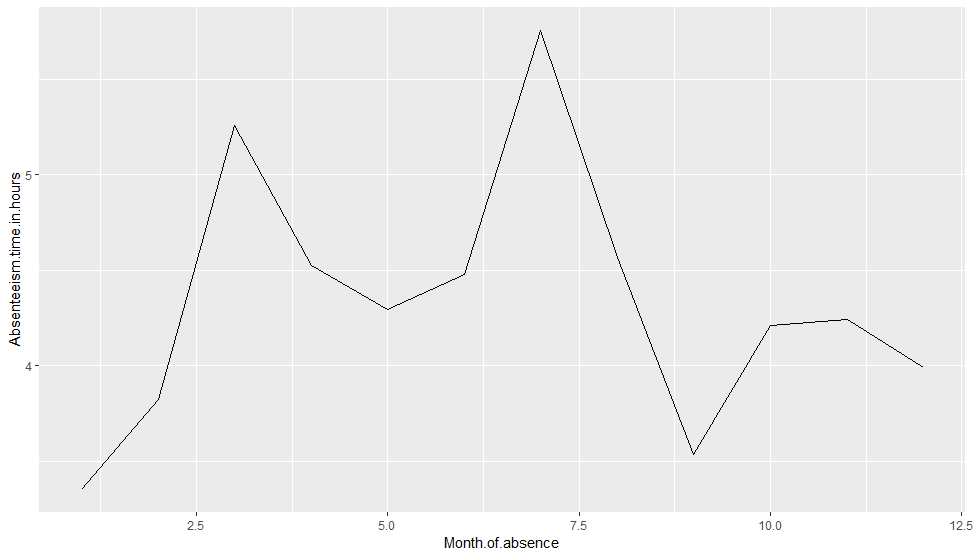


Fig 2.9 Line Plot of Absenteeism per month

From the above line plots, we clearly see the trend being formed in the data with respect to the month of the reading. The first plot shows the Monthly Utilization, which is a calculated field result of the difference in Service time and Absenteeism time whole divided by the Service Time.

For converting to time series, we filtered based on the month and took the mean of those observations to form the new observation for the month. While forming the new observation, we can ignore the variables that are just specific to each individual like Reason for absence can be different for each individual and would change as per individual and thus will not add any information that can help us forecast. Similarly, for the Height, Weight, Pet, and Son variables, so we ignore these variables from our forecast model as they are exogenous variables and cannot be aggregated for the full month.

All other variables are considered and provided to the Vector Auto Regression Model and the values for the next month are forecasted.

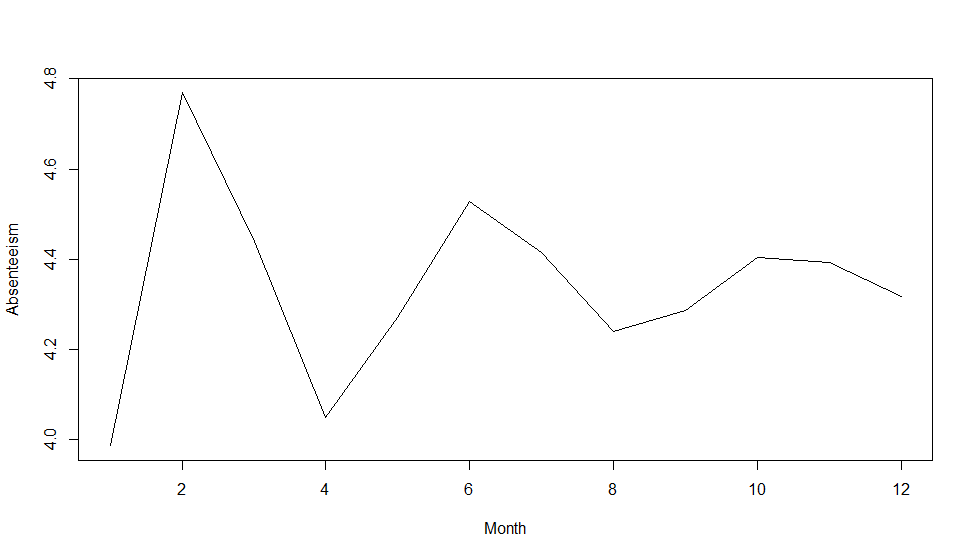


Fig 2.9 Forecasted Trend of Absenteeism

**Chapter 3**

**Conclusion**

**3.1 Model Evaluation**

Now that we have a few models for predicting the target variable, we need to evaluate and compare the models. We can compare the models using any of the following criteria:

1. Predictive Performance

2. Interpretability

3. Computational Eﬃciency

Predictive performance can be measured by comparing Predictions of the models with real values of the target variables, and calculating the error metrics for Regression.

The error metrics that we’ll be taking into consideration are:

* R-Square
  + R-Squared = 1 - (Explained Variation / Total Variation)
* Mean Square Error
  + the average squared difference between the estimated values and what is estimated.
* Mean Absolute Error
  + measures the average magnitude of the errors in a set of predictions, without considering their direction
* Explained Variance
  + measures the proportion to which a mathematical model accounts for the variation (dispersion) of a given data set.



Fig 3.1 Linear Models with their Error Metrics

**3.2 Answer to Problem Statement**

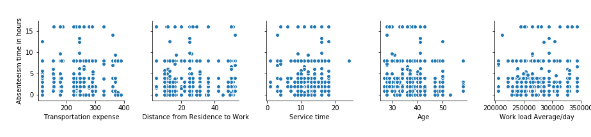
After thoroughly analyzing the data, we come to the conclusion that absenteeism is dependent on the following variables:

* Distance from Residence to Work
* Transportation Expense

So the changes company should bring to reduce the absenteeism time is that they can provide transportation to the employees with longer distances at reduced prices.

Whereas when we forecast the loss to occur, if the same trend of absenteeism continues, then from the Fig 2.9, we see the projected trend and the overall loss of utilization will be approx. 35% time wasted in absenteeism.

**Appendix A - Extra Figures**



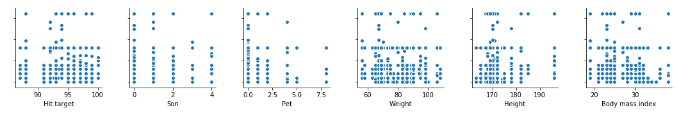
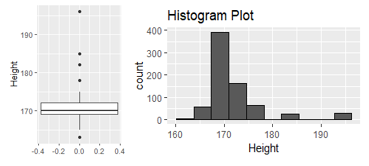
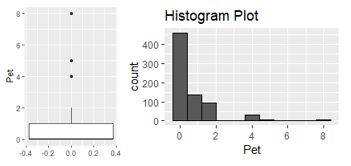
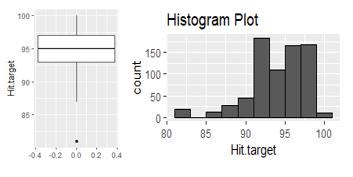
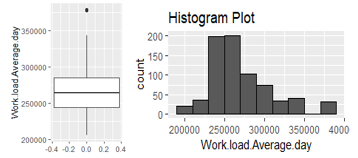
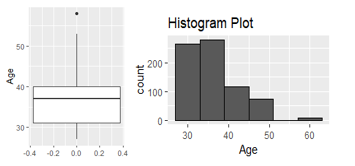


Fig 4.1 Scatter Plots for all the Numerical Variables w.r.t. Count



**Appendix B - Python Code**

import os

import pandas as pd

import numpy as np

import statistics as stat

import fancyimpute

import matplotlib.pyplot as plt

import seaborn as sns

from statsmodels.tsa.vector\_ar.var\_model import VAR

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

def load\_data():

# import the dataset

df=pd.read\_excel('Absenteeism\_at\_work\_Project.xls')

# Convert the categorical columns to object

cat\_cols=["ID", "Reason for absence", "Month of absence",

"Day of the week", "Seasons","Disciplinary failure",

"Education","Social drinker","Social smoker"]

for i in cat\_cols:

df[i] = df[i].astype(object)

num\_cols=["Transportation expense","Distance from Residence to Work", "Service time","Age",

"Work load Average/day ","Hit target","Son",

"Pet", "Weight","Height",

"Body mass index","Absenteeism time in hours" ]

for i in num\_cols:

df[i] = df[i].astype(np.float64)

# viewing the dataframe's info

df.info()

return df,num\_cols,cat\_cols

def impute\_missing\_vals(df,num\_cols,cat\_cols):

df\_full=pd.DataFrame(columns=df.columns)

for j in np.unique(df.ID):

df\_n=df[df.ID==j].reset\_index(drop=True)

missing\_val = df\_n.isnull().sum()

r,c=df\_n.shape

if r>2:

if df\_n[num\_cols].isnull().sum().sum()>0:

df\_n[num\_cols]=pd.DataFrame(fancyimpute.KNN(k = 5).complete(df\_n[num\_cols]), columns = num\_cols)

# for i in num\_cols:

# if len(df\_n[df\_n[i].isnull()])>0:

# df\_n.loc[df\_n[i].isnull(),i]=np.mean(df\_n[i])

for i in cat\_cols:

if len(df\_n[df\_n[i].isnull()])>0:

if len(stat.\_counts(df\_n.loc[:,i]))>1:

df\_n.loc[:,i]=df\_n.loc[:,i].fillna(method='ffill')

df\_n[i] = df\_n[i].astype(object)

else:

df\_n.loc[df\_n[i].isnull(),i]=stat.mode(df\_n[i])

df\_full=pd.concat([df\_full,df\_n],ignore\_index=True)

return df\_full

def outlier\_imputer(df\_o,num\_cols):

# Outlier Analysis

while True:

for i in num\_cols:

median=np.median(df\_o[i])

std=np.std(df\_o[i])

min=(df\_o[i].quantile(0.25)-1.5\*(df\_o[i].quantile(0.75)-df\_o[i].quantile(0.25)))

max=(df\_o[i].quantile(0.75)+1.5\*(df\_o[i].quantile(0.75)-df\_o[i].quantile(0.25)))

df\_o.loc[df\_o[i]<min,i] = np.nan

df\_o.loc[df\_o[i]>max,i] = np.nan

missing\_val = df\_o.isnull().sum()

print(missing\_val)

if(missing\_val.sum()>0):

df\_o[num\_cols]=pd.DataFrame(fancyimpute.KNN(k = 3).complete(df\_o[num\_cols]), columns = num\_cols)

# for i in num\_cols:

# if len(df\_o[df\_o[i].isnull()])>0:

# df\_o.loc[df\_o[i].isnull(),i]=np.mean(df\_o[i])

else:

break

df\_o.loc[df\_o['Reason for absence']==0,'Reason for absence']=20

df\_o.loc[df\_o['Month of absence']==0,'Month of absence']=stat.mode(df\_o['Month of absence'])

return df\_o

def add\_features(df):

# Feature Engineering

df['Time Utilization']=(df['Service time'] - df['Absenteeism time in hours'])/df['Service time']

for i in df['Month of absence'].unique():

monthly\_utilization=df[df['Month of absence']==i].iloc[:,21].sum()/len(df[df['Month of absence']==i])

print(monthly\_utilization)

if ('Monthly Utilization' in df.columns):

df.loc[df['Month of absence']==i,'Monthly Utilization']=monthly\_utilization

else:

df['Monthly Utilization']=np.where(df['Month of absence']==i,monthly\_utilization,np.nan)

return df

def convert\_to\_timeseries(df):

ts\_df=pd.DataFrame(columns=df.columns)

cat\_cols=['Day of the week', 'Disciplinary failure', 'Education','Reason for absence', 'Seasons', 'Social drinker','Social smoker']

num\_cols=["Transportation expense","Distance from Residence to Work", "Service time","Age",'Monthly Utilization',

"Work load Average/day ","Hit target","Son",

"Pet", "Weight","Height",

"Body mass index","Absenteeism time in hours" ]

for i in df['Month of absence'].unique():

x=df[df['Month of absence']==i]

n=pd.DataFrame(np.mean(x[num\_cols]).values.reshape(1,-1),columns=num\_cols)

c=pd.DataFrame(columns=cat\_cols)

for i in cat\_cols:

if len(stat.\_counts(x.loc[:,i]))>1:

c.loc[0,i]=stat.\_counts(x.loc[:,i])[0][0]

else:

c.loc[0,i]=stat.mode(x[i])

c['Month of absence']=np.mean(x['Month of absence'])

# s=pd.DataFrame(np.mean(df[df['Month of absence']==i]).values.reshape(1,-1),columns=df.columns)

ts\_df=pd.concat([ts\_df,pd.concat([n,c],axis=1,sort=True)],ignore\_index=True,sort=True)

ts\_df=ts\_df.sort\_values(by='Month of absence')

ts\_df.reset\_index(drop=True,inplace=True)

ts\_df.drop(labels=['ID','Time Utilization'],axis=1,inplace=True)

cat\_cols=['Day of the week', 'Disciplinary failure', 'Education','Reason for absence', 'Seasons', 'Social drinker','Social smoker']

for i in cat\_cols:

# ts\_df[i]=round(ts\_df[i],0)

ts\_df[i]=ts\_df[i].astype(object)

ts=ts\_df.set\_index(keys='Month of absence')

ts['Social smoker']=ts['Social smoker'].astype(np.float64)

ts['Social drinker']=ts['Social drinker'].astype(np.float64)

ts=pd.get\_dummies(ts,drop\_first=True)

# print(ts.head())

ts=ts.drop(columns=[

'Day of the week\_3', 'Day of the week\_4', 'Day of the week\_5',

'Day of the week\_6', 'Seasons\_2', 'Seasons\_3', 'Seasons\_4'],axis=1)

return ts

def feature\_selection(df,num\_cols):

#Set the width and hieght of the plot

f, ax = plt.subplots(figsize=(7, 5))

#Generate correlation matrix

corr = df[num\_cols].corr()

#Plot using seaborn library

sns.heatmap(corr, mask=np.zeros\_like(corr, dtype=np.bool), cmap=sns.diverging\_palette(220, 10, as\_cmap=True),

square=True, ax=ax)

return df

def train\_VAR(df):

endog=['Absenteeism time in hours', 'Distance from Residence to Work',

'Hit target', 'Service time',

'Transportation expense', 'Work load Average/day ']

exog=['Height','Monthly Utilization', 'Pet','Son','Weight', 'Reason for absence\_27.0', 'Reason for absence\_28.0']

#fit the model

model = VAR(endog=df[endog])

model\_fit = model.fit()

return model\_fit

def forecast(train,model\_fit):

# make prediction on validation

endog=['Absenteeism time in hours', 'Distance from Residence to Work',

'Hit target', 'Service time',

'Transportation expense', 'Work load Average/day ']

prediction = model\_fit.forecast(model\_fit.y, steps=12)

prediction=pd.DataFrame(prediction)

prediction.columns=train[endog].columns

prediction['Time Utilization']=(prediction['Service time'] - prediction['Absenteeism time in hours'])/prediction['Service time']

print('Total loss due to absenteeism ',1-np.mean(prediction['Time Utilization']),'%')

return prediction

def feature\_scaling(X\_train,X\_test):

standardScaler=StandardScaler()

X\_train[:,6:]=standardScaler.fit\_transform(X\_train[:,6:])

X\_test[:,6:]=standardScaler.transform(X\_test[:,6:])

return X\_train,X\_test,standardScaler

def train\_lm(df):

X=df.iloc[:,:20].values

y=df.iloc[:,20].values

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2,random\_state=123)

X\_train\_scaled,X\_test\_scaled,standardScaler=feature\_scaling(X\_train,X\_test)

lr\_model=LinearRegression()

lr\_model.fit(X\_train,y\_train)

print('Linear Model Accuracy ',lr\_model.score(X\_test,y\_test))

preds=lr\_model.predict(X\_test)

return pred

df,num\_cols,cat\_cols=load\_data()

df=impute\_missing\_vals(df,num\_cols,cat\_cols)

df=outlier\_imputer(df,["Service time","Work load Average/day ","Hit target","Absenteeism time in hours" ])

df=add\_features(df)

ts=convert\_to\_timeseries(df)

ts=feature\_selection(ts,ts.columns)

ts.drop(columns=["Body mass index","Age"],axis=1,inplace=True)

ts=feature\_selection(ts,ts.columns)

model=train\_VAR(ts)

pred=forecast(ts,model)

train\_lm(df

)**Appendix C - R Code**

library(readxl)

library(imputeMissings)

library(DMwR2)

library(psych)

library(ggplot2)

library(corrgram)

library(car)

library(dummies)

library(randomForest)

library(caTools)

library(FNN)

library(DMwR)

library(rpart)

library(forecast)

library(vars)

df = read\_excel('Absenteeism\_at\_work\_Project.xls')

df$`Reason for absence`[df$`Reason for absence`==0]=20

df$`Month of absence`[df$`Month of absence`==0]=NA

cat\_cols=c(1,2,3,4,5,13)

num\_cols=c(6:11,14,17:21)

for (i in cat\_cols) {

df[,i]=as.factor(df[,i,drop=T])

}

summary(df)

# Missing Value Imputation

missing\_val\_imputer=function(df,num\_cols,cat\_cols){

df\_full=df[0,]

for (i in unique(df$ID)) {

df\_n=df[df$ID==i,]

df\_n=impute(df\_n)

df\_full=rbind(df\_full,df\_n)

}

return(df\_full)

}

df=missing\_val\_imputer(df,num\_cols,cat\_cols)

# Lets step into visualization

# For Numerical Variables, we will be using the multi.hist fucntion to visualize

# All variables in one go. The plots will contain each variable's histogram,

# KDE plot, and a line representing Normal Distribution for comparison.

# Import package from library

multi.hist(df[,num\_cols[1:6]],dcol =c('blue','red'), dlty = c('solid','solid'),main = 'Variable Analysis' )

multi.hist(df[,num\_cols[-c(1:6)]],dcol =c('blue','red'), dlty = c('solid','solid'),main = 'Variable Analysis' )

# Now for the factors, lets plot bar graph to see the count of each class

init=ggplot(data=df)

for(i in cat\_cols){

plot=init+geom\_bar(aes(x=df[,colnames(df[,i,drop=F])]),fill='blue',colour='blue')+xlab(colnames(df[,i,drop=F]))+

ylab('Frequency')

print(plot)

}

# Plots the boxplots for all the numerical variables

for(i in num\_cols){

plot=init+geom\_boxplot(aes(y=df[,colnames(df[,i,drop=F])]))+ylab(colnames(df[,i,drop=F]))

print(plot)

}

# Creates the histogram for the variable that were found to be skewed

# lh is the vector containing index of the skewed variables

lh <- c(6,8,9,10,11,17,19,21)

# bw is binwidth to be selected for each variable

bw <- c(40,3,6,20000,2, 0.8, 3.6,12)

for (i in 1:8) {

plot <- init +

geom\_histogram(aes(x = df[,lh[i]]), binwidth = bw[i])+

geom\_vline(aes(xintercept = mean(df[,lh[i]])), color = "red")+

geom\_vline(aes(xintercept = median(df[,lh[i]])),color='blue')+

xlab(colnames(df)[lh[i]])+

ylab("Frequency")+

ggtitle(paste("Histogram, Median, and Mean of ",colnames(df)[lh[i]], ""))

print(plot)

}

# Creates the Histograms for the skewed Variables with Outliers

# lh is the vector containing index of the skewed variables

lh <- c(8,21)

# bw is binwidth to be selected for each variable

bw <- c(3,12)

for (i in 1:2) {

print( init+

geom\_histogram(aes(x = df[,lh[i]]), binwidth = bw[i],color='black')+

xlab(colnames(df)[lh[i]])+

ggtitle(paste("Histogram Plot")))

}

outlierImputer=function(df,num\_cols){

while (TRUE) {

tot\_miss=NULL

for(i in num\_cols){

val = df[,i][df[,i] %in% boxplot.stats(df[,i])$out]

df[,i][df[,i] %in% val]= NA

tot\_miss=c(tot\_miss,length(val))

}

print(sum(tot\_miss))

if(sum(tot\_miss)>0){

df=knnImputation(df,k = 3)

}

else{

break

}

}

return(df)

}

df=outlierImputer(df,c(8,21))

#Feature Engineering

df$Total.Utilization=(df$Service.time-df$Absenteeism.time.in.hours)/df$Service.time

for (i in unique(df$Month.of.absence)){

monthly\_utilization=sum(df[df$Month.of.absence==i,22])/length(df[df$Month.of.absence==i,22])

df[df$Month.of.absence==i,'Monthly.Utilization']=monthly\_utilization

}

# feature selection

numFeatureSel=function(df,num\_cols){

corr=cor(df[,num\_cols])

print('Eigen Values for Correlation Matrix')

print(eigen(corr)$values) # if values in decreasing order than multicollinearity present

# Condition Number: max Eigen Value / min Eigen Value

print('Condition Number')

print(max(eigen(corr)$values)/min(eigen(corr)$values)) # if grater than 100, multicollinearity present

corrgram(df[,num\_cols],order = F,upper.panel = panel.pie,text.panel = panel.txt,main='Correlation Plot')

return(num\_cols[-c(4,11)])

}

numSel=numFeatureSel(df,c(num\_cols))

numFeatureSel(df,num\_cols[-c(4,11)])

catFeatureSel=function(df,cat\_cols){

cSel=NULL

for(i in cat\_cols){

x=summary(aov(Absenteeism.time.in.hours~df[,i], data = df))

print(x)

if(x[[1]]$`Pr(>F)`[1]<0.05){

cSel=c(cSel,i)

}

}

return(cSel)

}

cSel=catFeatureSel(df,c(cat\_cols,12,16,15))

df\_o=df[,c(cSel[-1],numSel)]

df\_o=dummy.data.frame(df\_o, sep = "." )

df\_o[,c(29,1)]=NULL

VIF\_check=function(df\_o){

while (T) {

lr\_model=lm(Absenteeism.time.in.hours~.,data = df\_o)

x=vif(lr\_model)

maxVar = max(x)

if (maxVar > 6){

j = which(x == maxVar)

df\_o = df\_o[, -j]

}

else{

break()

}

}

return(df\_o)

}

df\_vif=VIF\_check(df\_o)

backwardEliminationRF=function(df,sl){

numVars = length(df)

for (i in c(1:numVars)){

imppred=randomForest(formula =Absenteeism.time.in.hours ~ ., data = df,ntree = 100, keep.forest = FALSE, importance = TRUE)

minVar =min(importance(imppred, type = 1))

if (minVar < sl){

j = which(importance(imppred, type = 1) == minVar)

df = df[, -j]

}

numVars = numVars - 1

}

return(imppred)

}

rf\_model=backwardEliminationRF(df\_o,5)

importance(rf\_model, type = 1)

backwardElimination=function(df,sl){

numVars = length(df)

for (i in c(1:numVars)){

regressor = lm(formula = Absenteeism.time.in.hours~., data = df)

maxVar = max(coef(summary(regressor))[c(2:numVars), "Pr(>|t|)"])

if (maxVar > sl){

j = which(coef(summary(regressor))[c(2:numVars), "Pr(>|t|)"] == maxVar)

df = df[, -j]

}

numVars = numVars - 1

}

return(regressor)

}

# Modeling the Data

# Divide data into train and test using stratified sampling method

set.seed(101)

sample = sample.split(df\_vif$Absenteeism.time.in.hours, SplitRatio = .80)

train = subset(df\_vif, sample == TRUE)

test = subset(df\_vif, sample == FALSE)

lr\_model=backwardElimination(df\_vif,0.05)

summary(lr\_model)

y\_pred=predict(lr\_model,test[,-c(1:3,7,11,15:16,22,26,28:36,38,40:44,46,47,48)])

#Summary of Linear model

summary(lr\_model)

r=sum((y\_pred-test[,48])^2)/sum((test[,48]-mean(test[,48]))^2)

1-r #-> 0.792

convertToTimeSeries=function(df){

ts=df[0,]

cat\_cols=c(2:5,12,13,14,15,16,17)

num\_cols=c(6:11,18:23)

for(m in unique(df$Month.of.absence)){

x=df[df$Month.of.absence==m,]

n=x[0,num\_cols]

n=rbind(n,data.frame(as.list(colMeans(x[,num\_cols]))))

c=x[0,]

for(i in cat\_cols){

m=names(table(x[,i]))[table(x[,i])==max(table(x[,i]))]

if(length(m)>1){

c[1,i]=m[1]

}

else{

c[1,i]=m[1]

}

}

c[1,num\_cols]=n

c[,c(12,14,15,16,17)]=as.numeric(c[,c(12,14,15,16,17)])

c=c[,-1]

ts=rbind(ts,c)

}

ts=ts[order(ts$Month.of.absence),]

rownames(ts)=ts$Month.of.absence

c[,cat\_cols]=as.numeric(c[,cat\_cols[-c(12,14,15,16,17)]])

c[,cat\_cols]=as.factor(c[,cat\_cols[-c(12,14,15,16,17)]])

ts$Month.of.absence=as.numeric(ts$Month.of.absence)

ts$Total.Utilization=NULL

return(ts)

}

ts=convertToTimeSeries(df)

ggplot(data = ts)+ geom\_line(aes(x=Month.of.absence,y=Monthly.Utilization))

ggplot(data = ts)+ geom\_line(aes(x=Month.of.absence,y=Absenteeism.time.in.hours))

ggplot(data = ts)+ geom\_line(aes(x=Work.load.Average.day,y=Absenteeism.time.in.hours))

ggplot(data = ts)+ geom\_line(aes(x=Work.load.Average.day,y=Monthly.Utilization))

ggplot(data = ts)+ geom\_line(aes(x=Distance.from.Residence.to.Work,y=Absenteeism.time.in.hours))

ggplot(data = ts)+ geom\_line(aes(x=Distance.from.Residence.to.Work,y=Monthly.Utilization))

ts\_full=stats::ts(ts[,c(5,6,7,9,10,20)])

var=VAR(y=ts\_full)

x=forecast(object = var,h=12)

ts\_pred=data.frame(x$forecast)

ts\_pred=ts\_pred[,c(1,6,11,16,21,26)]

colnames(ts\_pred)=colnames(ts[,c(5,6,7,9,10,20)])

ts\_pred$Total.Utilization=(ts\_pred$Service.time-ts\_pred$Absenteeism.time.in.hours)/ts\_pred$Service.time

print(paste('Total loss due to Absenteeism is ',(1-mean(ts\_pred$Total.Utilization)),'%'))

plot(ts\_pred$Absenteeism.time.in.hours,type='l',xlab='Month',ylab='Absenteeism')

**References**

Mohd Zubair 2016, ‘Predicting Wine Quality Hamza’

<https://www.statmethods.net/>

<https://www.rdocumentation.org/>

<https://www.kaggle.com/>