MiniProject

2023-12-02

Team :

- Rohith Ganni
- Pranthi Cavuturu
- Akhil Addepalli

Given Information:

Dataset Source : https://archive.ics.uci.edu/ml/datasets/Bias+correction+of+numerical+prediction+model+temperature+forecast

Target Variable to Predict : Next_Tmax

(a, b, c) Preprocessing, Splitting, Initial Model

Summary of the dataset is as follows:

```
library(readr)

## Warning: package 'readr' was built under R version 4.3.2

data <- read_csv("Bias_correction_ucl.csv")

## Rows: 7752 Columns: 25
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## dbl  (24): station, Present_Tmax, Present_Tmin, LDAPS_RHmin, LDAPS_RHmax,
LD...
## date  (1): Date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.

summary(data)

##     station        Date             Present_Tmax    Present_Tmin
##  Min.   : 1   Min.   :2013-06-30   Min.   :20.00   Min.   :11.30
##  1st Qu.: 7   1st Qu.:2014-07-15   1st Qu.:27.80   1st Qu.:21.70
##  Median :13   Median :2015-07-30   Median :29.90   Median :23.40
```

```
##  Mean   :13   Mean   :2015-07-30   Mean   :29.77   Mean   :23.23
##  3rd Qu.:19   3rd Qu.:2016-08-15   3rd Qu.:32.00   3rd Qu.:24.90
##  Max.   :25   Max.   :2017-08-30   Max.   :37.60   Max.   :29.90
##  NA's   :2    NA's   :2            NA's   :70      NA's   :70
##   LDAPS_RHmin     LDAPS_RHmax     LDAPS_Tmax_lapse LDAPS_Tmin_lapse
##  Min.   :19.79   Min.   : 58.94   Min.   :17.62    Min.   :14.27
##  1st Qu.:45.96   1st Qu.: 84.22   1st Qu.:27.67    1st Qu.:22.09
##  Median :55.04   Median : 89.79   Median :29.70    Median :23.76
##  Mean   :56.76   Mean   : 88.37   Mean   :29.61    Mean   :23.51
##  3rd Qu.:67.19   3rd Qu.: 93.74   3rd Qu.:31.71    3rd Qu.:25.15
##  Max.   :98.52   Max.   :100.00   Max.   :38.54    Max.   :29.62
##  NA's   :75      NA's   :75       NA's   :75       NA's   :75
##     LDAPS_WS         LDAPS_LH        LDAPS_CC1        LDAPS_CC2
##  Min.   : 2.883   Min.   :-13.60   Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 5.679   1st Qu.: 37.27   1st Qu.:0.1467   1st Qu.:0.1406
##  Median : 6.547   Median : 56.87   Median :0.3157   Median :0.3124
##  Mean   : 7.098   Mean   : 62.51   Mean   :0.3688   Mean   :0.3561
##  3rd Qu.: 8.032   3rd Qu.: 84.22   3rd Qu.:0.5755   3rd Qu.:0.5587
##  Max.   :21.858   Max.   :213.41   Max.   :0.9673   Max.   :0.9684
##  NA's   :75       NA's   :75       NA's   :75       NA's   :75
##    LDAPS_CC3        LDAPS_CC4         LDAPS_PPT1         LDAPS_PPT2
##  Min.   :0.0000   Min.   :0.00000   Min.   : 0.00000   Min.   : 0.00000
##  1st Qu.:0.1014   1st Qu.:0.08153   1st Qu.: 0.00000   1st Qu.: 0.00000
##  Median :0.2626   Median :0.22766   Median : 0.00000   Median : 0.00000
##  Mean   :0.3184   Mean   :0.29919   Mean   : 0.59199   Mean   : 0.48500
##  3rd Qu.:0.4967   3rd Qu.:0.49949   3rd Qu.: 0.05252   3rd Qu.: 0.01836
##  Max.   :0.9838   Max.   :0.97471   Max.   :23.70154   Max.   :21.62166
##  NA's   :75       NA's   :75        NA's   :75         NA's   :75
##    LDAPS_PPT3        LDAPS_PPT4            lat             lon
##  Min.   : 0.0000   Min.   : 0.00000   Min.   :37.46   Min.   :126.8
##  1st Qu.: 0.0000   1st Qu.: 0.00000   1st Qu.:37.51   1st Qu.:126.9
##  Median : 0.0000   Median : 0.00000   Median :37.55   Median :127.0
##  Mean   : 0.2782   Mean   : 0.26941   Mean   :37.54   Mean   :127.0
##  3rd Qu.: 0.0079   3rd Qu.: 0.00004   3rd Qu.:37.58   3rd Qu.:127.0
##  Max.   :15.8412   Max.   :16.65547   Max.   :37.65   Max.   :127.1
##  NA's   :75        NA's   :75
##       DEM             Slope         Solar radiation   Next_Tmax
##  Min.   : 12.37   Min.   :0.09847   Min.   :4330    Min.   :17.40
##  1st Qu.: 28.70   1st Qu.:0.27130   1st Qu.:4999    1st Qu.:28.20
##  Median : 45.72   Median :0.61800   Median :5436    Median :30.50
##  Mean   : 61.87   Mean   :1.25705   Mean   :5342    Mean   :30.27
##  3rd Qu.: 59.83   3rd Qu.:1.76780   3rd Qu.:5728    3rd Qu.:32.60
##  Max.   :212.34   Max.   :5.17823   Max.   :5993    Max.   :38.90
##                                                     NA's   :27
##    Next_Tmin
##  Min.   :11.30
##  1st Qu.:21.30
##  Median :23.10
##  Mean   :22.93
##  3rd Qu.:24.60
```

```
##  Max.   :29.80
##  NA's   :27
```

Moving on with three different approaches of handling NULL Values:

1. Imputing the missing values with the mean values of the variables
2. Imputing the missing values with the median values of the variables
3. Omitting the Null Values

Common step in all the above methods is to remove the NULL values in the 'date' column

```
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.3.2

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

initial_structure <- dim(data)

data <- data %>% filter(!is.na(Date))

cleaned_structure <- dim(data)

print(initial_structure)

## [1] 7752   25

print(cleaned_structure)

## [1] 7750   25

data_med <- data
data_mean <- data
```

Approach 1 - Imputing with mean values

```
# Imputing missing values with mean for numeric columns
numeric_columns <- sapply(data_mean, is.numeric)
data_mean[numeric_columns] <- lapply(data_mean[numeric_columns], function(x)
ifelse(is.na(x), mean(x, na.rm = TRUE), x))

data_mean$station <- as.factor(data_mean$station)
```

```r
str(data_mean)
```

```
## spc_tbl_ [7,750 × 25] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ station         : Factor w/ 25 levels "1","2","3","4",..: 1 2 3 4 5 6 7
8 9 10 ...
##  $ Date            : Date[1:7750], format: "2013-06-30" "2013-06-30" ...
##  $ Present_Tmax    : num [1:7750] 28.7 31.9 31.6 32 31.4 31.9 31.4 32.1
31.4 31.6 ...
##  $ Present_Tmin    : num [1:7750] 21.4 21.6 23.3 23.4 21.9 23.5 24.4 23.6
22 20.5 ...
##  $ LDAPS_RHmin     : num [1:7750] 58.3 52.3 48.7 58.2 56.2 ...
##  $ LDAPS_RHmax     : num [1:7750] 91.1 90.6 84 96.5 90.2 ...
##  $ LDAPS_Tmax_lapse: num [1:7750] 28.1 29.9 30.1 29.7 29.1 ...
##  $ LDAPS_Tmin_lapse: num [1:7750] 23 24 24.6 23.3 23.5 ...
##  $ LDAPS_WS        : num [1:7750] 6.82 5.69 6.14 5.65 5.74 ...
##  $ LDAPS_LH        : num [1:7750] 69.5 51.9 20.6 65.7 108 ...
##  $ LDAPS_CC1       : num [1:7750] 0.234 0.226 0.209 0.216 0.151 ...
##  $ LDAPS_CC2       : num [1:7750] 0.204 0.252 0.257 0.226 0.25 ...
##  $ LDAPS_CC3       : num [1:7750] 0.162 0.159 0.204 0.161 0.179 ...
##  $ LDAPS_CC4       : num [1:7750] 0.131 0.128 0.142 0.134 0.17 ...
##  $ LDAPS_PPT1      : num [1:7750] 0 0 0 0 0 0 0 0 0 0 ...
##  $ LDAPS_PPT2      : num [1:7750] 0 0 0 0 0 0 0 0 0 0 ...
##  $ LDAPS_PPT3      : num [1:7750] 0 0 0 0 0 0 0 0 0 0 ...
##  $ LDAPS_PPT4      : num [1:7750] 0 0 0 0 0 0 0 0 0 0 ...
##  $ lat             : num [1:7750] 37.6 37.6 37.6 37.6 37.6 ...
##  $ lon             : num [1:7750] 127 127 127 127 127 ...
##  $ DEM             : num [1:7750] 212.3 44.8 33.3 45.7 35 ...
##  $ Slope           : num [1:7750] 2.785 0.514 0.266 2.535 0.505 ...
##  $ Solar radiation : num [1:7750] 5993 5869 5864 5857 5860 ...
##  $ Next_Tmax       : num [1:7750] 29.1 30.5 31.1 31.7 31.2 31.5 30.9 31.1
31.3 30.5 ...
##  $ Next_Tmin       : num [1:7750] 21.2 22.5 23.9 24.3 22.5 24 23.4 22.9
21.6 21 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   station = col_double(),
##   ..   Date = col_date(format = ""),
##   ..   Present_Tmax = col_double(),
##   ..   Present_Tmin = col_double(),
##   ..   LDAPS_RHmin = col_double(),
##   ..   LDAPS_RHmax = col_double(),
##   ..   LDAPS_Tmax_lapse = col_double(),
##   ..   LDAPS_Tmin_lapse = col_double(),
##   ..   LDAPS_WS = col_double(),
##   ..   LDAPS_LH = col_double(),
##   ..   LDAPS_CC1 = col_double(),
##   ..   LDAPS_CC2 = col_double(),
##   ..   LDAPS_CC3 = col_double(),
##   ..   LDAPS_CC4 = col_double(),
```

```
##   ..   LDAPS_PPT1 = col_double(),
##   ..   LDAPS_PPT2 = col_double(),
##   ..   LDAPS_PPT3 = col_double(),
##   ..   LDAPS_PPT4 = col_double(),
##   ..   lat = col_double(),
##   ..   lon = col_double(),
##   ..   DEM = col_double(),
##   ..   Slope = col_double(),
##   ..   `Solar radiation` = col_double(),
##   ..   Next_Tmax = col_double(),
##   ..   Next_Tmin = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

After Imputing the Values with mean, we split this data set into train, validation and test data sets. Following is the number of rows for the entire data set, train, vaidation & test data sets respectively

```
data_mean <- data_mean[order(data_mean$Date), ]

train_size_mean <- round(nrow(data_mean) * 0.60)
valid_size_mean <- round(nrow(data_mean) * 0.80)

train_data_mean <- data_mean[1:train_size_mean, ]
valid_data_mean <- data_mean[(train_size_mean + 1):valid_size_mean, ]
test_data_mean <- data_mean[(valid_size_mean + 1):nrow(data_mean), ]

nrow(data_mean)

## [1] 7750

nrow(train_data_mean)

## [1] 4650

nrow(valid_data_mean)

## [1] 1550

nrow(test_data_mean)

## [1] 1550
```

Now we apply regression model and train it with the train dataset for mean and check it's Evaluation Metric (Root Mean Squared Error [RMSE]) value for our Approach - 1

```
predictors <- setdiff(names(train_data_mean), c('Next_Tmax', 'Date',
'station', 'Next_Tmin'))
train_data_subset_mean <- train_data_mean[, c('Next_Tmax', predictors)]

model_mean <- lm(Next_Tmax ~ ., data = train_data_subset_mean)
valid_data_subset_mean <- valid_data_mean[, predictors]
```

```r
predictions_mean <- predict(model_mean, newdata = valid_data_subset_mean)

rmse <- sqrt(mean((valid_data_mean$Next_Tmax - predictions_mean)^2))
rmse
```

```
## [1] 1.644064
```

Approach 2 - Imputing with median values

Structure of the data set when imputed with median values is as follows:

```r
# Imputing missing values with median for numeric columns
numeric_columns <- sapply(data_med, is.numeric)
data_med[numeric_columns] <- lapply(data_med[numeric_columns], function(x)
ifelse(is.na(x), median(x, na.rm = TRUE), x))

data_med$station <- as.factor(data_med$station)

str(data_med)
```

```
## spc_tbl_ [7,750 × 25] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ station         : Factor w/ 25 levels "1","2","3","4",..: 1 2 3 4 5 6 7
8 9 10 ...
##  $ Date            : Date[1:7750], format: "2013-06-30" "2013-06-30" ...
##  $ Present_Tmax    : num [1:7750] 28.7 31.9 31.6 32 31.4 31.9 31.4 32.1
31.4 31.6 ...
##  $ Present_Tmin    : num [1:7750] 21.4 21.6 23.3 23.4 21.9 23.5 24.4 23.6
22 20.5 ...
##  $ LDAPS_RHmin     : num [1:7750] 58.3 52.3 48.7 58.2 56.2 ...
##  $ LDAPS_RHmax     : num [1:7750] 91.1 90.6 84 96.5 90.2 ...
##  $ LDAPS_Tmax_lapse: num [1:7750] 28.1 29.9 30.1 29.7 29.1 ...
##  $ LDAPS_Tmin_lapse: num [1:7750] 23 24 24.6 23.3 23.5 ...
##  $ LDAPS_WS        : num [1:7750] 6.82 5.69 6.14 5.65 5.74 ...
##  $ LDAPS_LH        : num [1:7750] 69.5 51.9 20.6 65.7 108 ...
##  $ LDAPS_CC1       : num [1:7750] 0.234 0.226 0.209 0.216 0.151 ...
##  $ LDAPS_CC2       : num [1:7750] 0.204 0.252 0.257 0.226 0.25 ...
##  $ LDAPS_CC3       : num [1:7750] 0.162 0.159 0.204 0.161 0.179 ...
##  $ LDAPS_CC4       : num [1:7750] 0.131 0.128 0.142 0.134 0.17 ...
##  $ LDAPS_PPT1      : num [1:7750] 0 0 0 0 0 0 0 0 0 0 ...
##  $ LDAPS_PPT2      : num [1:7750] 0 0 0 0 0 0 0 0 0 0 ...
##  $ LDAPS_PPT3      : num [1:7750] 0 0 0 0 0 0 0 0 0 0 ...
##  $ LDAPS_PPT4      : num [1:7750] 0 0 0 0 0 0 0 0 0 0 ...
##  $ lat             : num [1:7750] 37.6 37.6 37.6 37.6 37.6 ...
##  $ lon             : num [1:7750] 127 127 127 127 127 ...
##  $ DEM             : num [1:7750] 212.3 44.8 33.3 45.7 35 ...
##  $ Slope           : num [1:7750] 2.785 0.514 0.266 2.535 0.505 ...
##  $ Solar radiation : num [1:7750] 5993 5869 5864 5857 5860 ...
##  $ Next_Tmax       : num [1:7750] 29.1 30.5 31.1 31.7 31.2 31.5 30.9 31.1
31.3 30.5 ...
##  $ Next_Tmin       : num [1:7750] 21.2 22.5 23.9 24.3 22.5 24 23.4 22.9
21.6 21 ...
```

```
##  - attr(*, "spec")=
##   .. cols(
##   ..   station = col_double(),
##   ..   Date = col_date(format = ""),
##   ..   Present_Tmax = col_double(),
##   ..   Present_Tmin = col_double(),
##   ..   LDAPS_RHmin = col_double(),
##   ..   LDAPS_RHmax = col_double(),
##   ..   LDAPS_Tmax_lapse = col_double(),
##   ..   LDAPS_Tmin_lapse = col_double(),
##   ..   LDAPS_WS = col_double(),
##   ..   LDAPS_LH = col_double(),
##   ..   LDAPS_CC1 = col_double(),
##   ..   LDAPS_CC2 = col_double(),
##   ..   LDAPS_CC3 = col_double(),
##   ..   LDAPS_CC4 = col_double(),
##   ..   LDAPS_PPT1 = col_double(),
##   ..   LDAPS_PPT2 = col_double(),
##   ..   LDAPS_PPT3 = col_double(),
##   ..   LDAPS_PPT4 = col_double(),
##   ..   lat = col_double(),
##   ..   lon = col_double(),
##   ..   DEM = col_double(),
##   ..   Slope = col_double(),
##   ..   `Solar radiation` = col_double(),
##   ..   Next_Tmax = col_double(),
##   ..   Next_Tmin = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

After Imputing the Values with median, we split this data set into train, validation and test data sets. Following is the number of rows for the entire data set, train, vaidation & test data sets respectively

```
data_med <- data_med[order(data_med$Date), ]

train_size_med <- round(nrow(data_med) * 0.60)
valid_size_med <- round(nrow(data_med) * 0.80)

train_data_med <- data_med[1:train_size_med, ]
valid_data_med <- data_med[(train_size_med + 1):valid_size_med, ]
test_data_med <- data_med[(valid_size_med + 1):nrow(data_med), ]

nrow(data_med)

## [1] 7750

nrow(train_data_med)

## [1] 4650
```

```
nrow(valid_data_med)

## [1] 1550

nrow(test_data_med)

## [1] 1550
```

Now we apply regression model and train it with the train dataset for median and check it's Evaluation Metric (Root Mean Squared Error [RMSE]) value for our Approach - 2

```
predictors <- setdiff(names(train_data_med), c('Next_Tmax', 'Date',
'station', 'Next_Tmin'))
train_data_subset_med <- train_data_med[, c('Next_Tmax', predictors)]

model_med <- lm(Next_Tmax ~ ., data = train_data_subset_med)
valid_data_subset_med <- valid_data_med[, predictors]
predictions_med <- predict(model_med, newdata = valid_data_subset_med)

rmse <- sqrt(mean((valid_data_med$Next_Tmax - predictions_med)^2))
rmse

## [1] 1.629672
```

Approach 3 - Omitting NULL values

```
data <- na.omit(data)
data$station <- as.factor(data$station)
sum(is.na(data))

## [1] 0

nrow(data)

## [1] 7588
```

After removing the NULL values, we split this data set into train, validation and test data sets. Following is the number of rows for the entire data set, train, vaidation & test data sets respectively

```
data <- data[order(data$Date), ]

train_size <- round(nrow(data) * 0.60)
valid_size <- round(nrow(data) * 0.80)

train_data <- data[1:train_size, ]
valid_data <- data[(train_size + 1):valid_size, ]
test_data <- data[(valid_size + 1):nrow(data), ]

nrow(data)

## [1] 7588
```

```
nrow(train_data)

## [1] 4553

nrow(valid_data)

## [1] 1517

nrow(test_data)

## [1] 1518
```

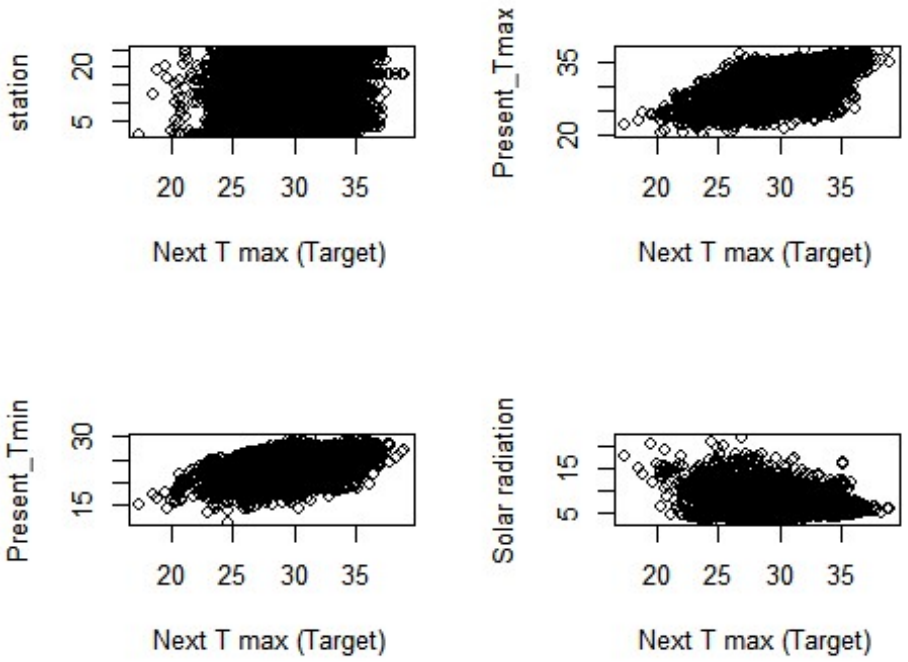
Now we apply regression model and train it with the train dataset and check it's Evaluation Metric (Root Mean Squared Error [RMSE]) value for our Approach - 3

```
predictors <- setdiff(names(train_data), c('Next_Tmax', 'Date', 'station',
'Next_Tmin'))
train_data_subset <- train_data[, c('Next_Tmax', predictors)]

model <- lm(Next_Tmax ~ ., data = train_data_subset)
valid_data_subset <- valid_data[, predictors]
predictions <- predict(model, newdata = valid_data_subset)

rmse <- sqrt(mean((valid_data$Next_Tmax - predictions)^2))
rmse

## [1] 1.513085
```

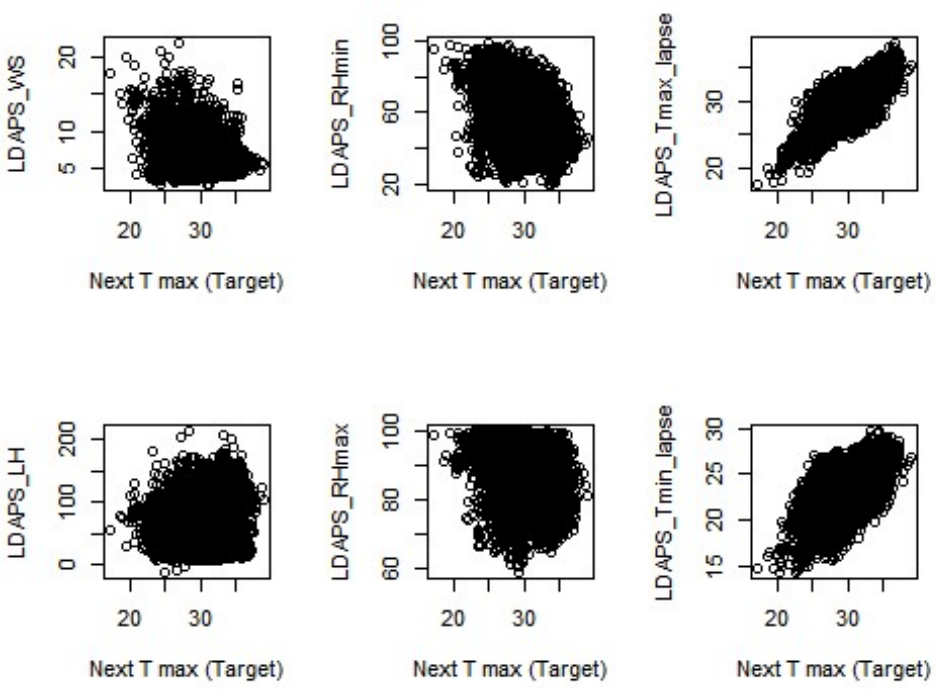
From the above three approaches, it is clearly evident that the Approach - 3 which omits null values is the best method of handling null values for the given data set with the least RMSE Value of 1.513. Thus, this approach is selected for further improvements

Target Features vs Different Variables in the Dataset

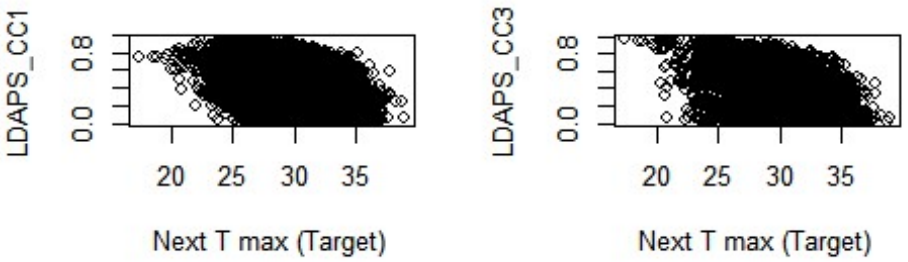
```
layout(matrix(1:4,2,2))
 plot(data$Next_Tmax,data$station , ylab="station", xlab="Next T max
(Target)")
 plot(data$Next_Tmax,data$Present_Tmin , ylab="Present_Tmin", xlab="Next T
max (Target)")
 plot(data$Next_Tmax,data$Present_Tmax , ylab="Present_Tmax", xlab="Next T
max (Target)")
 plot(data$Next_Tmax,data$LDAPS_WS , ylab="Solar radiation", xlab="Next T max
(Target)")
```



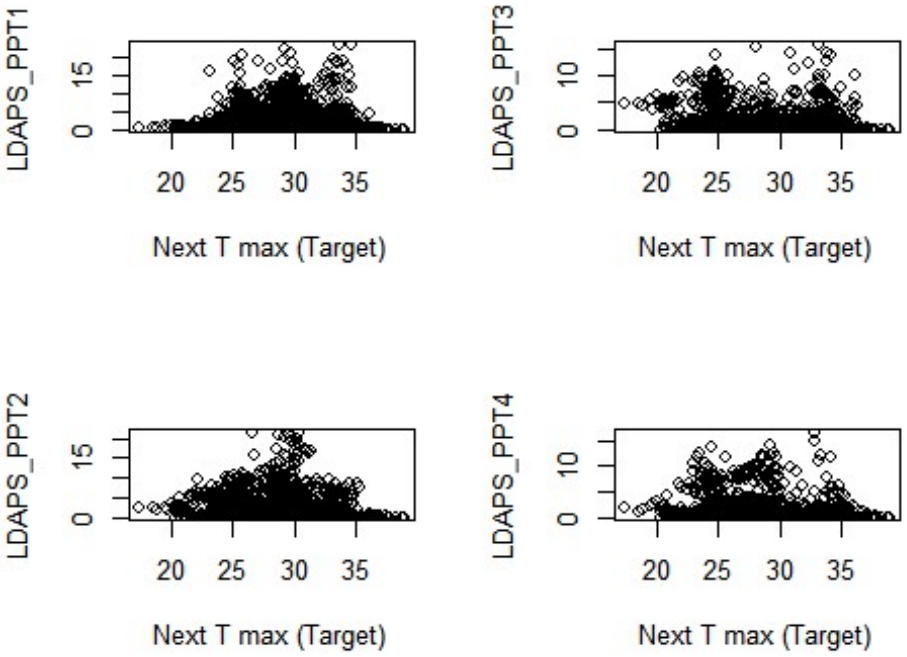

```
layout(matrix(1:6,2,3))
 plot(data$Next_Tmax,data$LDAPS_WS , ylab="LDAPS_WS", xlab="Next T max
(Target)")
 plot(data$Next_Tmax,data$LDAPS_LH , ylab="LDAPS_LH", xlab="Next T max
(Target)")
 plot(data$Next_Tmax,data$LDAPS_RHmin , ylab="LDAPS_RHmin", xlab="Next T max
(Target)")
 plot(data$Next_Tmax,data$LDAPS_RHmax , ylab="LDAPS_RHmax", xlab="Next T max
(Target)")
 plot(data$Next_Tmax,data$LDAPS_Tmax_lapse , ylab="LDAPS_Tmax_lapse",
xlab="Next T max (Target)")
 plot(data$Next_Tmax,data$LDAPS_Tmin_lapse , ylab="LDAPS_Tmin_lapse",
xlab="Next T max (Target)")
```



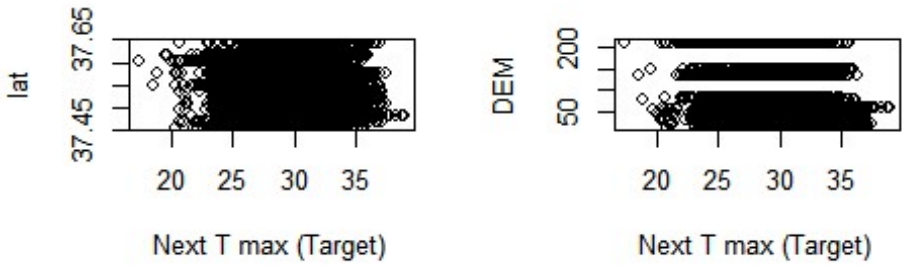

```
layout(matrix(1:4,2,2))
 plot(data$Next_Tmax,data$LDAPS_CC1 , ylab="LDAPS_CC1", xlab="Next T max
(Target)")
 plot(data$Next_Tmax,data$LDAPS_CC2 , ylab="LDAPS_CC2", xlab="Next T max
(Target)")
 plot(data$Next_Tmax,data$LDAPS_CC3 , ylab="LDAPS_CC3", xlab="Next T max
(Target)")
 plot(data$Next_Tmax,data$LDAPS_CC4 , ylab="LDAPS_CC4", xlab="Next T max
(Target)")
```

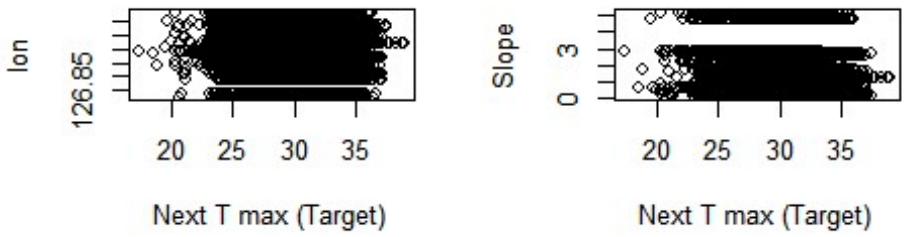



```
layout(matrix(1:4,2,2))
 plot(data$Next_Tmax,data$LDAPS_PPT1 , ylab="LDAPS_PPT1", xlab="Next T max
(Target)")
 plot(data$Next_Tmax,data$LDAPS_PPT2 , ylab="LDAPS_PPT2", xlab="Next T max
(Target)")
 plot(data$Next_Tmax,data$LDAPS_PPT3 , ylab="LDAPS_PPT3", xlab="Next T max
(Target)")
 plot(data$Next_Tmax,data$LDAPS_PPT4 , ylab="LDAPS_PPT4", xlab="Next T max
(Target)")
```

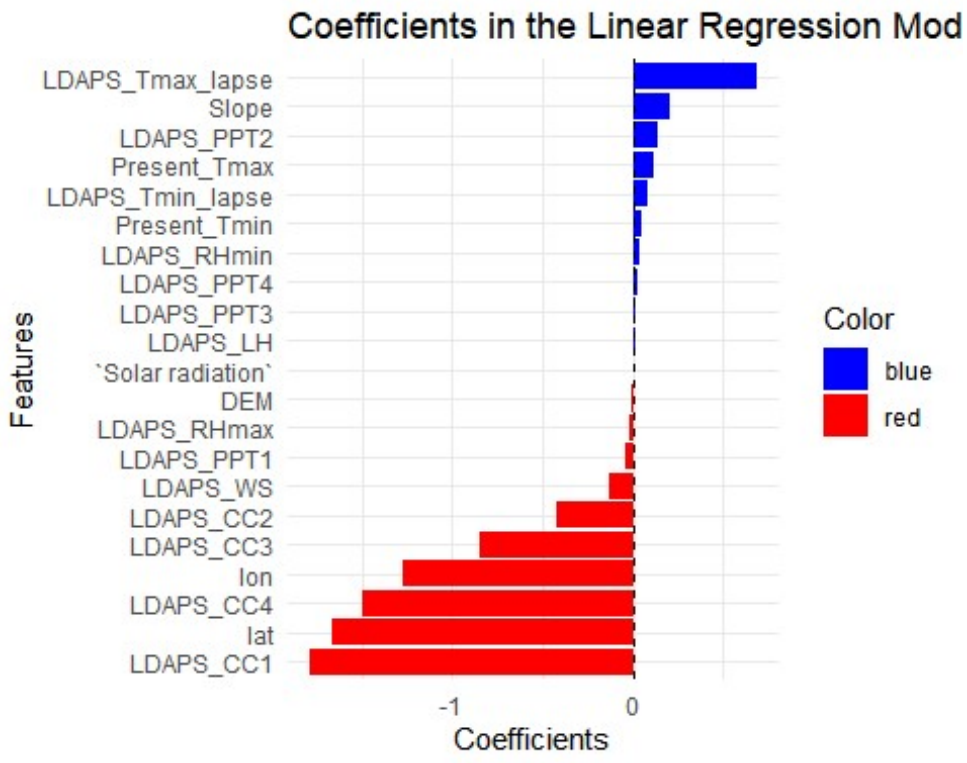



```r
layout(matrix(1:4,2,2))
plot(data$Next_Tmax,data$lat , ylab="lat", xlab="Next T max (Target)")
plot(data$Next_Tmax,data$lon , ylab="lon", xlab="Next T max (Target)")
plot(data$Next_Tmax,data$DEM , ylab="DEM", xlab="Next T max (Target)")
plot(data$Next_Tmax,data$Slope , ylab="Slope", xlab="Next T max (Target)")
```

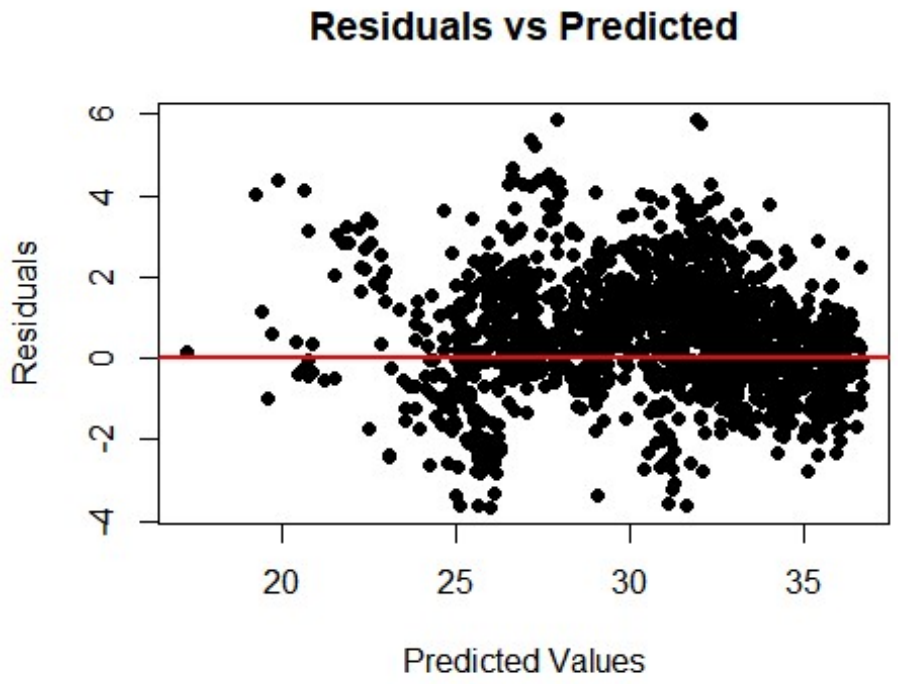





```r
coefficients <- coef(model)[-1]  # Exclude the intercept

coefficients_df <- data.frame(Feature = names(coefficients), Coefficient =
coefficients)
coefficients_df$Color <- ifelse(coefficients_df$Coefficient > 0, "blue",
"red")

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.2

ggplot(coefficients_df, aes(x = reorder(Feature, Coefficient), y =
Coefficient, fill = Color)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_minimal() +
  labs(x = "Features", y = "Coefficients", title = "Coefficients in the
Linear Regression Model") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +
  scale_fill_manual(values = c("blue", "red"))
```




```r
# Residuals Plot
residuals <- valid_data$Next_Tmax - predictions

plot(predictions, residuals,
     xlab = "Predicted Values",
     ylab = "Residuals",
     main = "Residuals vs Predicted",
     pch = 19,
     col = "black")
abline(h = 0, col = "red", lwd = 2)
```



(d) Improved Model

Model 1

We calculate the P-value every variable in the dataset to know what are the "non-significant" variables for analysis. By "non-significant", here we mean which variables doesn't contribute much to predict the target variable

P-Value > 0.05 is taken into consideration to consider variables to be non-significant under the following conditions for Hypothesis

H_0 : Coefficient of Variable $\beta_i = 0$ H_1 : Coefficient of Variable $\beta_i \neq 0$

Thus if P-Value > 0.05, it means that the coefficient of that particular variable in the multiple linear regression equation is 0 i.e. predicted value is not dependent on this variable

For our Approach - 3, let's move on to find such "non-significant" variables

```
summary(model)$coefficients[, "Pr(>|t|)"]
```

```
##       (Intercept)      Present_Tmax      Present_Tmin       LDAPS_RHmin
##      6.060376e-11      9.644519e-21      7.675173e-03      8.812238e-25
##       LDAPS_RHmax  LDAPS_Tmax_lapse  LDAPS_Tmin_lapse          LDAPS_WS
##      4.527696e-08     2.250477e-231      1.764642e-03      9.290196e-36
##          LDAPS_LH         LDAPS_CC1         LDAPS_CC2         LDAPS_CC3
##      3.173148e-27      3.062570e-31      2.677528e-02      7.754436e-06
```

```
##         LDAPS_CC4        LDAPS_PPT1        LDAPS_PPT2        LDAPS_PPT3
##      7.492563e-24      6.134319e-05      2.698886e-24      5.557203e-01
##        LDAPS_PPT4               lat               lon               DEM
##      2.598408e-01      2.266374e-04      6.829667e-06      1.716810e-12
##             Slope `Solar radiation`
##      2.223459e-15      1.353044e-02
```

```
non_significant_vars <- summary(model)$coefficients[, "Pr(>|t|)"] > 0.05
names(non_significant_vars[non_significant_vars])
```

```
## [1] "LDAPS_PPT3" "LDAPS_PPT4"
```

From the above result, “LDAPS_PPT3” “LDAPS_PPT4” are the “non-significant” variables. Thus, we remove them and apply the model again to test the RMSE value. Along with the above variables, we also removed redundant variables (Eg: Station code can be used instead of lat, lon, DEM, Slope)

```
# Remove the non-significant columns and Next_Tmin from the dataset
data$LDAPS_PPT3 <- NULL
data$LDAPS_PPT4 <- NULL
data$DEM <- NULL
data$Next_Tmin <- NULL

train_df1 <- subset(train_data, select = -c(Next_Tmin, Date, lon, lat, Slope,
DEM, LDAPS_PPT3, LDAPS_PPT4))
valid_df1 <- subset(valid_data, select = -c(Next_Tmax, Next_Tmin, Date, lon,
lat, Slope, DEM, LDAPS_PPT3, LDAPS_PPT4))

model1 <- lm(Next_Tmax ~ ., data = train_df1)

summary(model1)
```

```
##
## Call:
## lm(formula = Next_Tmax ~ ., data = train_df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5277 -0.8230  0.0042  0.8172  5.2315
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        5.036e+00  5.784e-01   8.708  < 2e-16 ***
## station2           8.512e-01  1.472e-01   5.782 7.86e-09 ***
## station3           7.122e-01  1.642e-01   4.338 1.47e-05 ***
## station4           1.704e+00  1.499e-01  11.363  < 2e-16 ***
## station5           9.943e-01  1.495e-01   6.652 3.23e-11 ***
## station6           1.171e+00  1.537e-01   7.622 3.02e-14 ***
## station7           1.125e+00  1.536e-01   7.324 2.83e-13 ***
## station8           1.219e+00  1.516e-01   8.042 1.12e-15 ***
```

```
## station9           1.656e+00  1.463e-01  11.322  < 2e-16 ***
## station10          1.460e+00  1.448e-01  10.084  < 2e-16 ***
## station11          1.192e+00  1.524e-01   7.819 6.58e-15 ***
## station12          1.260e+00  1.552e-01   8.113 6.30e-16 ***
## station13          1.109e+00  1.590e-01   6.972 3.58e-12 ***
## station14          1.195e+00  1.624e-01   7.361 2.16e-13 ***
## station15          9.804e-01  1.562e-01   6.277 3.77e-10 ***
## station16          6.100e-01  1.447e-01   4.215 2.55e-05 ***
## station17          8.210e-01  1.500e-01   5.475 4.62e-08 ***
## station18          2.368e+00  1.504e-01  15.741  < 2e-16 ***
## station19          1.121e+00  1.508e-01   7.431 1.28e-13 ***
## station20          2.240e+00  1.469e-01  15.244  < 2e-16 ***
## station21          4.741e-01  1.633e-01   2.904 0.003703 **
## station22          1.234e+00  1.491e-01   8.274  < 2e-16 ***
## station23          1.832e+00  1.494e-01  12.264  < 2e-16 ***
## station24          1.411e+00  1.543e-01   9.146  < 2e-16 ***
## station25          1.322e+00  1.627e-01   8.121 5.90e-16 ***
## Present_Tmax       7.358e-02  1.183e-02   6.222 5.37e-10 ***
## Present_Tmin      -3.623e-03  1.605e-02  -0.226 0.821469
## LDAPS_RHmin        3.110e-02  3.672e-03   8.470  < 2e-16 ***
## LDAPS_RHmax       -2.499e-02  4.143e-03  -6.032 1.75e-09 ***
## LDAPS_Tmax_lapse   6.777e-01  1.892e-02  35.819  < 2e-16 ***
## LDAPS_Tmin_lapse   1.483e-01  2.384e-02   6.219 5.46e-10 ***
## LDAPS_WS          -1.236e-01  1.067e-02 -11.577  < 2e-16 ***
## LDAPS_LH           7.268e-03  1.209e-03   6.010 2.01e-09 ***
## LDAPS_CC1         -1.847e+00  1.488e-01 -12.408  < 2e-16 ***
## LDAPS_CC2         -3.618e-01  1.800e-01  -2.010 0.044468 *
## LDAPS_CC3         -6.629e-01  1.800e-01  -3.684 0.000232 ***
## LDAPS_CC4         -1.476e+00  1.368e-01 -10.787  < 2e-16 ***
## LDAPS_PPT1        -3.985e-02  1.134e-02  -3.515 0.000445 ***
## LDAPS_PPT2         1.499e-01  1.302e-02  11.520  < 2e-16 ***
## `Solar radiation`  1.033e-04  5.303e-05   1.947 0.051592 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.348 on 4513 degrees of freedom
## Multiple R-squared:  0.7613, Adjusted R-squared:  0.7592
## F-statistic: 369.1 on 39 and 4513 DF,  p-value: < 2.2e-16

predictions1 <- predict(model1, newdata = valid_df1)

# Calculate new RMSE
rmse1 <- sqrt(mean((valid_data$Next_Tmax - predictions1)^2))
rmse1

## [1] 1.465021

anova(model1, model)

## Analysis of Variance Table
##
```

```
## Model 1: Next_Tmax ~ station + Present_Tmax + Present_Tmin + LDAPS_RHmin +
##     LDAPS_RHmax + LDAPS_Tmax_lapse + LDAPS_Tmin_lapse + LDAPS_WS +
##     LDAPS_LH + LDAPS_CC1 + LDAPS_CC2 + LDAPS_CC3 + LDAPS_CC4 +
##     LDAPS_PPT1 + LDAPS_PPT2 + `Solar radiation`
## Model 2: Next_Tmax ~ Present_Tmax + Present_Tmin + LDAPS_RHmin +
LDAPS_RHmax +
##     LDAPS_Tmax_lapse + LDAPS_Tmin_lapse + LDAPS_WS + LDAPS_LH +
##     LDAPS_CC1 + LDAPS_CC2 + LDAPS_CC3 + LDAPS_CC4 + LDAPS_PPT1 +
##     LDAPS_PPT2 + LDAPS_PPT3 + LDAPS_PPT4 + lat + lon + DEM +
##     Slope + `Solar radiation`
##   Res.Df    RSS  Df Sum of Sq      F    Pr(>F)
## 1   4513 8205.3
## 2   4531 8984.2 -18   -778.98 23.803 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From Anova above we can conclude that RSS value has decreased. Hence, Model1 is better fit for this data than Model.

Model 2

In Model2 to avoid over fitting the model with Cloud coverage and Precipitation values, we simplify the data by adding new columns “Night_cloud_cover”, “Day_Cloud_Cover”, “Precipitation” and removed respective old columns.

```
train_df2 <- within(train_df1, {
  Night_Cloud_Cover = (LDAPS_CC1 + LDAPS_CC4) / 2
  Day_Cloud_Cover = (LDAPS_CC2 + LDAPS_CC3) / 2
  Precipitation = (LDAPS_PPT1 + LDAPS_PPT2) / 2

  # Remove the original columns
  LDAPS_CC1 = NULL
  LDAPS_CC2 = NULL
  LDAPS_CC3 = NULL
  LDAPS_CC4 = NULL
  LDAPS_PPT1 = NULL
  LDAPS_PPT2 = NULL
})

head(train_df2)
```

```
## # A tibble: 6 × 14
##   station Present_Tmax Present_Tmin LDAPS_RHmin LDAPS_RHmax
LDAPS_Tmax_lapse
##   <fct>          <dbl>        <dbl>       <dbl>       <dbl>
<dbl>
## 1 1               28.7         21.4        58.3        91.1
28.1
## 2 2               31.9         21.6        52.3        90.6
29.9
```

```
## 3 3               31.6         23.3        48.7        84.0
30.1
## 4 4               32           23.4        58.2        96.5
29.7
## 5 5               31.4         21.9        56.2        90.2
29.1
## 6 6               31.9         23.5        52.4        85.3
29.2
## # ℹ 8 more variables: LDAPS_Tmin_lapse <dbl>, LDAPS_WS <dbl>, LDAPS_LH
<dbl>,
## #   `Solar radiation` <dbl>, Next_Tmax <dbl>, Precipitation <dbl>,
## #   Day_Cloud_Cover <dbl>, Night_Cloud_Cover <dbl>
```

```
valid_df2 <- within(valid_df1, {
  Night_Cloud_Cover = (LDAPS_CC1 + LDAPS_CC4) / 2
  Day_Cloud_Cover = (LDAPS_CC2 + LDAPS_CC3) / 2
  Precipitation = LDAPS_PPT1 + LDAPS_PPT2 / 2

  # Remove the original columns
  LDAPS_CC1 = NULL
  LDAPS_CC2 = NULL
  LDAPS_CC3 = NULL
  LDAPS_CC4 = NULL
  LDAPS_PPT1 = NULL
  LDAPS_PPT2 = NULL
})
head(valid_df2)
```

```
## # A tibble: 6 × 13
##   station Present_Tmax Present_Tmin LDAPS_RHmin LDAPS_RHmax
LDAPS_Tmax_lapse
##   <fct>          <dbl>        <dbl>       <dbl>       <dbl>
<dbl>
## 1 14              30.6         19.8        33.8        76.5
30.6
## 2 15              30.4         19.9        35.8        77.7
30.3
## 3 16              28.8         18          37.4        90.0
29.4
## 4 17              28.9         16.5        39.8        94.2
29.9
## 5 18              30.6         19.9        36.4        90.3
29.8
## 6 19              30.7         19.4        35.5        79.6
29.9
## # ℹ 7 more variables: LDAPS_Tmin_lapse <dbl>, LDAPS_WS <dbl>, LDAPS_LH
<dbl>,
## #   `Solar radiation` <dbl>, Precipitation <dbl>, Day_Cloud_Cover <dbl>,
## #   Night_Cloud_Cover <dbl>
```

```r
model2 <- lm(Next_Tmax ~ ., data = train_df2)

predictions2 <- predict(model2, newdata = valid_df2)

rmse <- sqrt(mean((valid_data$Next_Tmax - predictions2)^2))

print(rmse)
```

```
## [1] 1.431952
```

```r
summary(model2)
```

```
##
## Call:
## lm(formula = Next_Tmax ~ ., data = train_df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4626 -0.8074  0.0237  0.8218  5.2902
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        5.358e+00  5.784e-01   9.264  < 2e-16 ***
## station2           8.619e-01  1.486e-01   5.800 7.08e-09 ***
## station3           7.296e-01  1.656e-01   4.407 1.07e-05 ***
## station4           1.726e+00  1.516e-01  11.387  < 2e-16 ***
## station5           1.016e+00  1.512e-01   6.718 2.07e-11 ***
## station6           1.188e+00  1.553e-01   7.654 2.37e-14 ***
## station7           1.147e+00  1.553e-01   7.384 1.82e-13 ***
## station8           1.222e+00  1.532e-01   7.974 1.93e-15 ***
## station9           1.667e+00  1.478e-01  11.275  < 2e-16 ***
## station10          1.445e+00  1.464e-01   9.874  < 2e-16 ***
## station11          1.214e+00  1.540e-01   7.884 3.93e-15 ***
## station12          1.257e+00  1.567e-01   8.027 1.26e-15 ***
## station13          1.130e+00  1.605e-01   7.043 2.17e-12 ***
## station14          1.189e+00  1.638e-01   7.255 4.70e-13 ***
## station15          9.908e-01  1.577e-01   6.283 3.64e-10 ***
## station16          6.012e-01  1.464e-01   4.107 4.09e-05 ***
## station17          8.241e-01  1.516e-01   5.436 5.73e-08 ***
## station18          2.380e+00  1.520e-01  15.653  < 2e-16 ***
## station19          1.119e+00  1.523e-01   7.351 2.32e-13 ***
## station20          2.250e+00  1.486e-01  15.139  < 2e-16 ***
## station21          5.014e-01  1.648e-01   3.043  0.00236 **
## station22          1.259e+00  1.508e-01   8.347  < 2e-16 ***
## station23          1.849e+00  1.510e-01  12.246  < 2e-16 ***
## station24          1.426e+00  1.559e-01   9.145  < 2e-16 ***
## station25          1.332e+00  1.642e-01   8.111 6.43e-16 ***
## Present_Tmax       8.439e-02  1.151e-02   7.332 2.67e-13 ***
## Present_Tmin      -2.217e-02  1.603e-02  -1.383  0.16685
## LDAPS_RHmin        3.409e-02  3.654e-03   9.329  < 2e-16 ***
```

```
## LDAPS_RHmax       -2.934e-02  4.046e-03  -7.251 4.83e-13 ***
## LDAPS_Tmax_lapse   6.835e-01  1.888e-02  36.201  < 2e-16 ***
## LDAPS_Tmin_lapse   1.367e-01  2.346e-02   5.828 6.02e-09 ***
## LDAPS_WS          -1.224e-01  1.079e-02 -11.347  < 2e-16 ***
## LDAPS_LH           7.178e-03  1.214e-03   5.912 3.62e-09 ***
## `Solar radiation`  1.086e-04  5.299e-05   2.050  0.04044 *
## Precipitation      8.416e-02  1.453e-02   5.793 7.39e-09 ***
## Day_Cloud_Cover   -5.374e-01  2.232e-01  -2.408  0.01608 *
## Night_Cloud_Cover -3.656e+00  2.038e-01 -17.935  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.365 on 4516 degrees of freedom
## Multiple R-squared:  0.7554, Adjusted R-squared:  0.7534
## F-statistic: 387.4 on 36 and 4516 DF,  p-value: < 2.2e-16
```

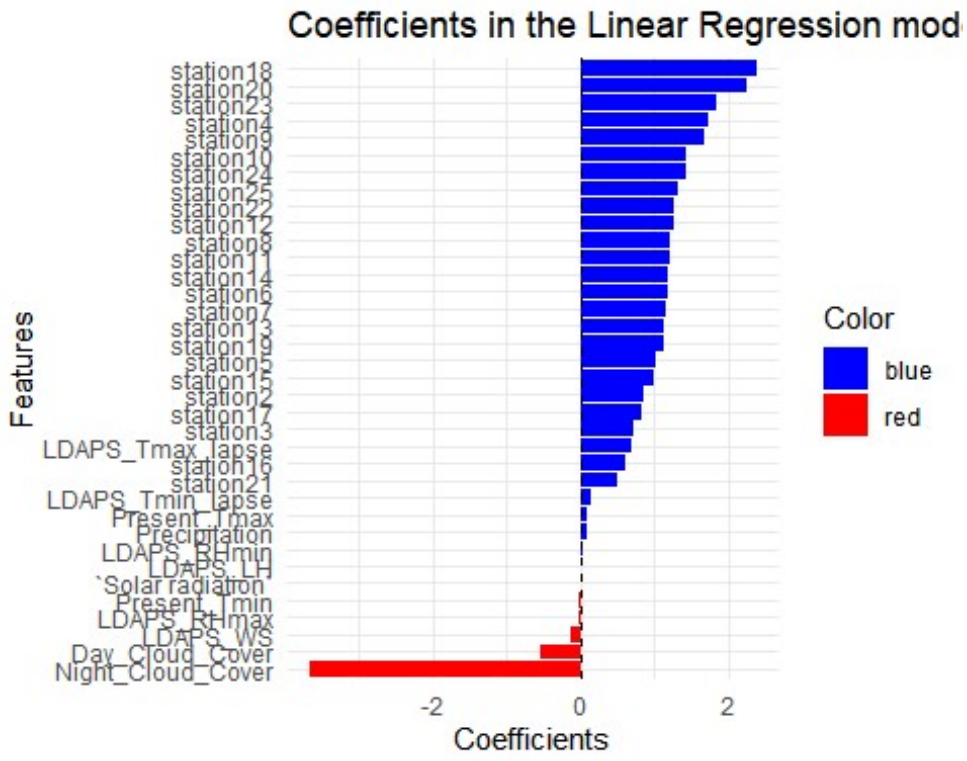
```
anova(model2, model1)
```

```
## Analysis of Variance Table
##
## Model 1: Next_Tmax ~ station + Present_Tmax + Present_Tmin + LDAPS_RHmin +
##     LDAPS_RHmax + LDAPS_Tmax_lapse + LDAPS_Tmin_lapse + LDAPS_WS +
##     LDAPS_LH + `Solar radiation` + Precipitation + Day_Cloud_Cover +
##     Night_Cloud_Cover
## Model 2: Next_Tmax ~ Present_Tmax + Present_Tmin + LDAPS_RHmin +
LDAPS_RHmax +
##     LDAPS_Tmax_lapse + LDAPS_Tmin_lapse + LDAPS_WS + LDAPS_LH +
##     LDAPS_CC1 + LDAPS_CC2 + LDAPS_CC3 + LDAPS_CC4 + LDAPS_PPT1 +
##     LDAPS_PPT2 + LDAPS_PPT3 + LDAPS_PPT4 + lat + lon + DEM +
##     Slope + `Solar radiation`
##   Res.Df    RSS  Df Sum of Sq      F    Pr(>F)
## 1   4516 8408.7
## 2   4531 8984.2 -15   -575.55 20.607 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

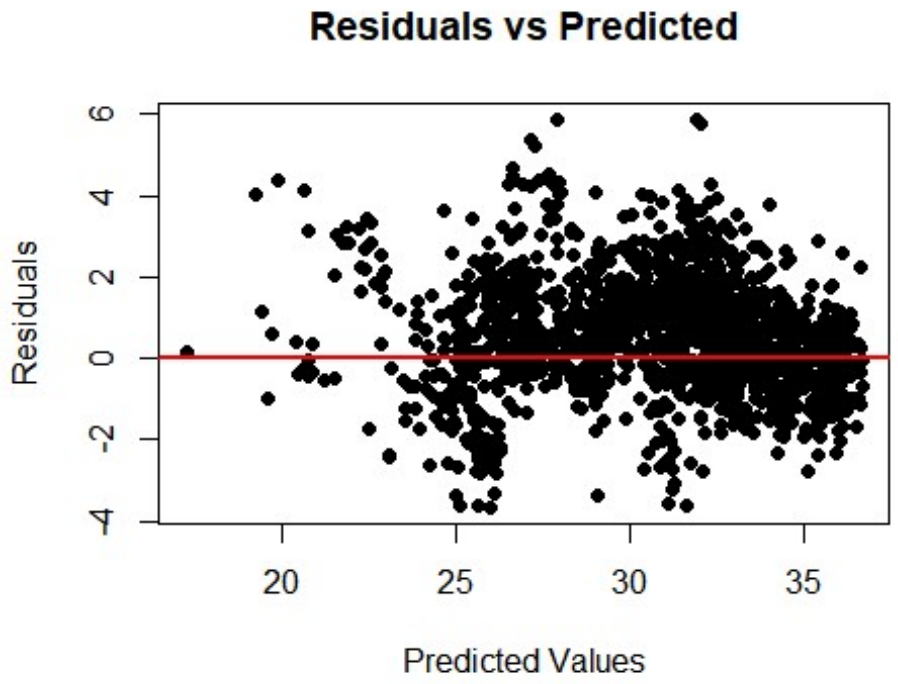
From above, we can conclude that RMSE value has decreased. Hence, Model2 is better fit for this data than Model1, which shows better predictive capability of model2.

```
coefficients <- coef(model2)[-1]

coefficients_df <- data.frame(Feature = names(coefficients), Coefficient =
coefficients)
coefficients_df$Color <- ifelse(coefficients_df$Coefficient > 0, "blue",
"red")
library(ggplot2)
ggplot(coefficients_df, aes(x = reorder(Feature, Coefficient), y =
Coefficient, fill = Color)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_minimal() +
```

```
  labs(x = "Features", y = "Coefficients", title = "Coefficients in the
Linear Regression model2") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +
  scale_fill_manual(values = c("blue", "red"))
```




```
residuals <- valid_data$Next_Tmax - predictions

plot(predictions, residuals,
     xlab = "Predicted Values",
     ylab = "Residuals",
     main = "Residuals vs Predicted",
     pch = 19,
     col = "black")
abline(h = 0, col = "red", lwd = 2)
```



Model 3

Now we again calculate p -values with newly added columns, and remove "non-significant" variables for analysis.

```
summary(model2)$coefficients[, "Pr(>|t|)"]
```

```
##       (Intercept)          station2          station3          station4
##      2.966230e-20      7.080116e-09      1.072493e-05      1.230780e-29
##          station5          station6          station7          station8
##      2.070290e-11      2.374584e-14      1.823000e-13      1.927438e-15
##          station9         station10         station11         station12
##      4.242386e-29      9.207510e-23      3.934371e-15      1.261330e-15
##         station13         station14         station15         station16
##      2.168611e-12      4.698410e-13      3.638506e-10      4.087312e-05
##         station17         station18         station19         station20
##      5.728521e-08      8.063978e-54      2.322741e-13      1.523495e-50
##         station21         station22         station23         station24
##      2.357054e-03      9.193862e-17      6.063895e-34      8.825244e-20
##         station25      Present_Tmax      Present_Tmin       LDAPS_RHmin
##      6.429237e-16      2.674240e-13      1.668512e-01      1.631955e-20
##       LDAPS_RHmax  LDAPS_Tmax_lapse  LDAPS_Tmin_lapse          LDAPS_WS
##      4.826490e-13     3.494525e-252      6.015211e-09      1.912460e-29
##          LDAPS_LH `Solar radiation`     Precipitation   Day_Cloud_Cover
##      3.624385e-09      4.044390e-02      7.386235e-09      1.608306e-02
## Night_Cloud_Cover
##      1.531597e-69
```

```r
non_significant_vars <- summary(model2)$coefficients[, "Pr(>|t|)"] > 0.05
names(non_significant_vars[non_significant_vars])
```

```
## [1] "Present_Tmin"
```

From the above result, “Present_Tmin” are the “non-significant” variables. Thus, we remove them and apply the model again to test the RMSE value.

```r
train_df3 = train_df2
valid_df3 = valid_df2

train_df3$Present_Tmin <- NULL
valid_df3$Present_Tmin <- NULL

model3 <- lm(Next_Tmax ~ ., data = train_df3)

predictions3 <- predict(model3, newdata = valid_df3)

rmse <- sqrt(mean((valid_data$Next_Tmax - predictions3)^2))

print(rmse)
```

```
## [1] 1.427989
```

```r
summary(model3)$coefficients[, "Pr(>|t|)"]
```

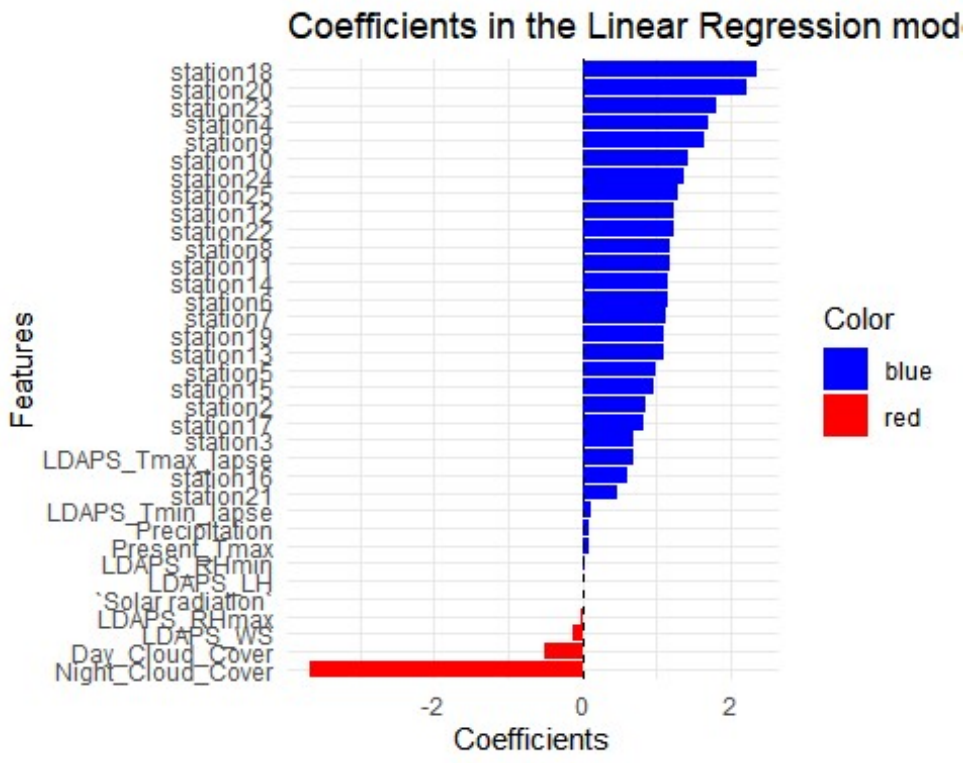
```
##       (Intercept)          station2          station3          station4
##      3.581151e-20      8.594860e-09      2.160638e-05      2.697849e-29
##          station5          station6          station7          station8
##      4.033351e-11      6.193104e-14      4.504811e-13      5.049505e-15
##          station9         station10         station11         station12
##      1.076020e-28      1.920073e-22      1.031099e-14      2.148432e-15
##         station13         station14         station15         station16
##      5.713142e-12      1.163389e-12      7.797537e-10      4.532234e-05
##         station17         station18         station19         station20
##      5.767960e-08      7.621044e-54      4.573071e-13      2.818758e-50
##         station21         station22         station23         station24
##      4.452705e-03      2.360320e-16      1.155000e-33      2.068781e-19
##         station25      Present_Tmax       LDAPS_RHmin       LDAPS_RHmax
##      1.676716e-15      6.580759e-13      2.214436e-20      3.348754e-14
##  LDAPS_Tmax_lapse  LDAPS_Tmin_lapse          LDAPS_WS          LDAPS_LH
##     8.432877e-253      8.531739e-09      1.839901e-30      6.035723e-09
## `Solar radiation`     Precipitation   Day_Cloud_Cover Night_Cloud_Cover
##      2.627857e-02      1.810371e-08      2.034394e-02      1.499942e-70
```

```r
non_significant_vars <- summary(model3)$coefficients[, "Pr(>|t|)"] > 0.05
names(non_significant_vars[non_significant_vars])
```

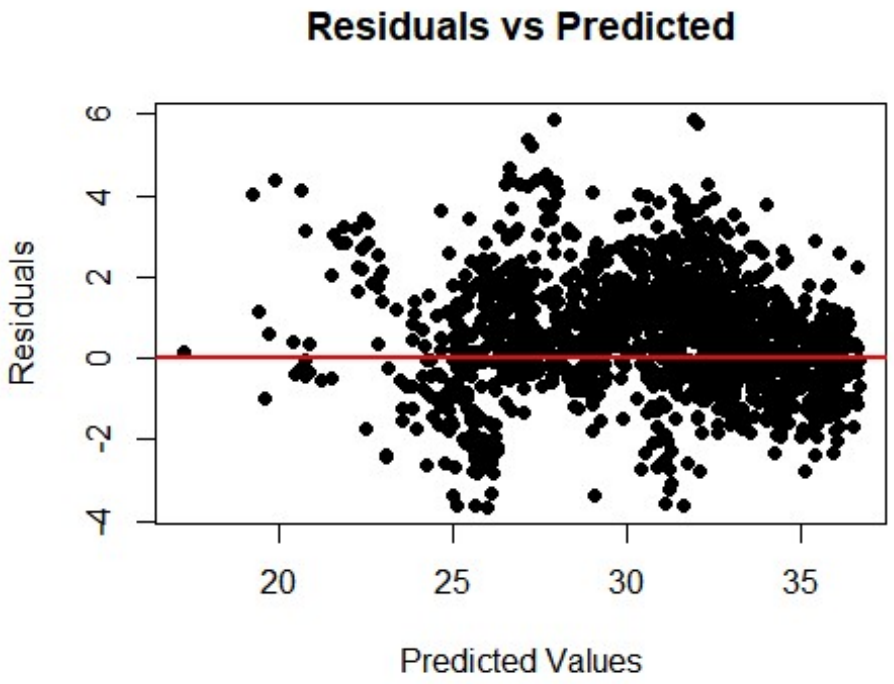
```
## character(0)
```

```r
anova(model3, model)
```

```
## Analysis of Variance Table
##
## Model 1: Next_Tmax ~ station + Present_Tmax + LDAPS_RHmin + LDAPS_RHmax +
##     LDAPS_Tmax_lapse + LDAPS_Tmin_lapse + LDAPS_WS + LDAPS_LH +
##     `Solar radiation` + Precipitation + Day_Cloud_Cover +
Night_Cloud_Cover
## Model 2: Next_Tmax ~ Present_Tmax + Present_Tmin + LDAPS_RHmin +
LDAPS_RHmax +
##     LDAPS_Tmax_lapse + LDAPS_Tmin_lapse + LDAPS_WS + LDAPS_LH +
##     LDAPS_CC1 + LDAPS_CC2 + LDAPS_CC3 + LDAPS_CC4 + LDAPS_PPT1 +
##     LDAPS_PPT2 + LDAPS_PPT3 + LDAPS_PPT4 + lat + lon + DEM +
##     Slope + `Solar radiation`
##   Res.Df    RSS  Df Sum of Sq      F    Pr(>F)
## 1   4517 8412.2
## 2   4531 8984.2 -14   -571.99 21.938 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

coefficients <- coef(model3)[-1]

coefficients_df <- data.frame(Feature = names(coefficients), Coefficient =
coefficients)
coefficients_df$Color <- ifelse(coefficients_df$Coefficient > 0, "blue",
"red")
library(ggplot2)
ggplot(coefficients_df, aes(x = reorder(Feature, Coefficient), y =
Coefficient, fill = Color)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_minimal() +
  labs(x = "Features", y = "Coefficients", title = "Coefficients in the
Linear Regression model2") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "black") +
  scale_fill_manual(values = c("blue", "red"))
```

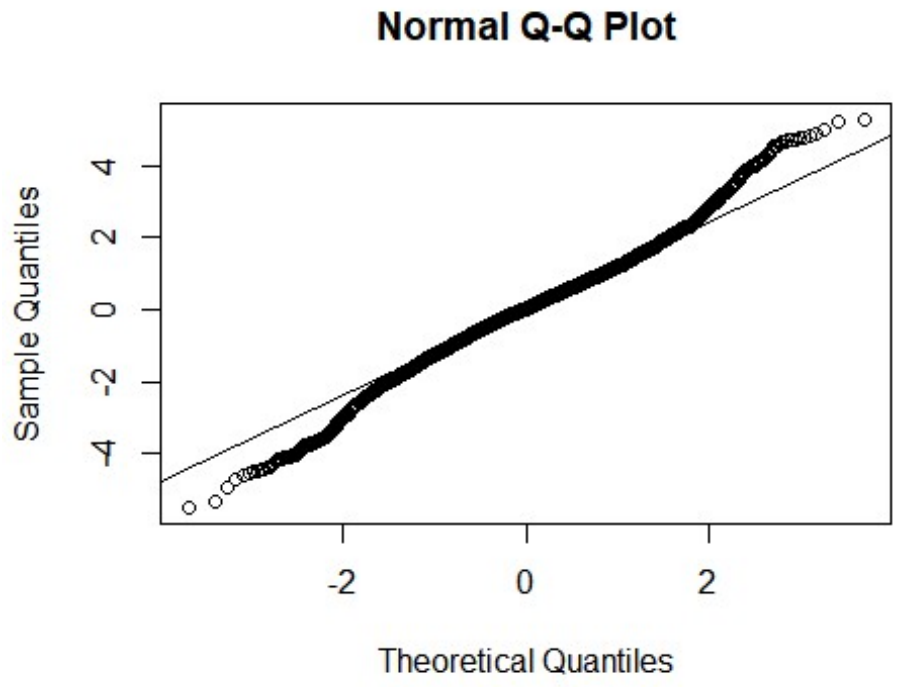



```r
# Residuals Plot
residuals <- valid_data$Next_Tmax - predictions

plot(predictions, residuals,
     xlab = "Predicted Values",
     ylab = "Residuals",
     main = "Residuals vs Predicted",
     pch = 19,
     col = "black")
abline(h = 0, col = "red", lwd = 2)
```



From above analysis we dont have any P-values that are > 0.05, but from above coefficients histogram we can tell that Next_Tmax is least dependent on LDAPS_LH, Solar Radiation, LDAPS_RHmax

```
shapiro.test(model3$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  model3$residuals
## W = 0.98899, p-value < 2.2e-16

{
 qqnorm(model3$residuals)
 qqline(model3$residuals)
}
```



The middle portion of the plot, where points conform more closely to a straight line, suggests that the data distribution is approximately normal.

Results from Test Data

```
test_data_subset <- test_data[, predictors]
test_predictions <- predict(model, newdata = test_data_subset)
test_rmse1 <- sqrt(mean((test_data$Next_Tmax - test_predictions)^2))
test_rmse1

## [1] 1.647331

test_df2 <- within(test_data, {
  Night_Cloud_Cover = (LDAPS_CC1 + LDAPS_CC4) / 2
  Day_Cloud_Cover = (LDAPS_CC2 + LDAPS_CC3) / 2
  Precipitation = (LDAPS_PPT1 + LDAPS_PPT2) / 2

  LDAPS_CC1 = NULL
  LDAPS_CC2 = NULL
  LDAPS_CC3 = NULL
  LDAPS_CC4 = NULL
  LDAPS_PPT1 = NULL
  LDAPS_PPT2 = NULL
  Next_Tmin = NULL
  Date = NULL
  lon = NULL
  lat = NULL
  Slope = NULL
```

```
  DEM = NULL
  LDAPS_PPT3 = NULL
  LDAPS_PPT4 = NULL
})

test_predictions2 <- predict(model3, newdata = test_df2)
test_rmse2 <- sqrt(mean((test_data$Next_Tmax - test_predictions2)^2))
test_rmse2
```

```
## [1] 1.591924
```

Based on the RMSE values comparision, it clearly shows that our Improved model can predict with better accuracy than the initial model.