Energy Forecasting using Multilinear Regression

Team 1 :

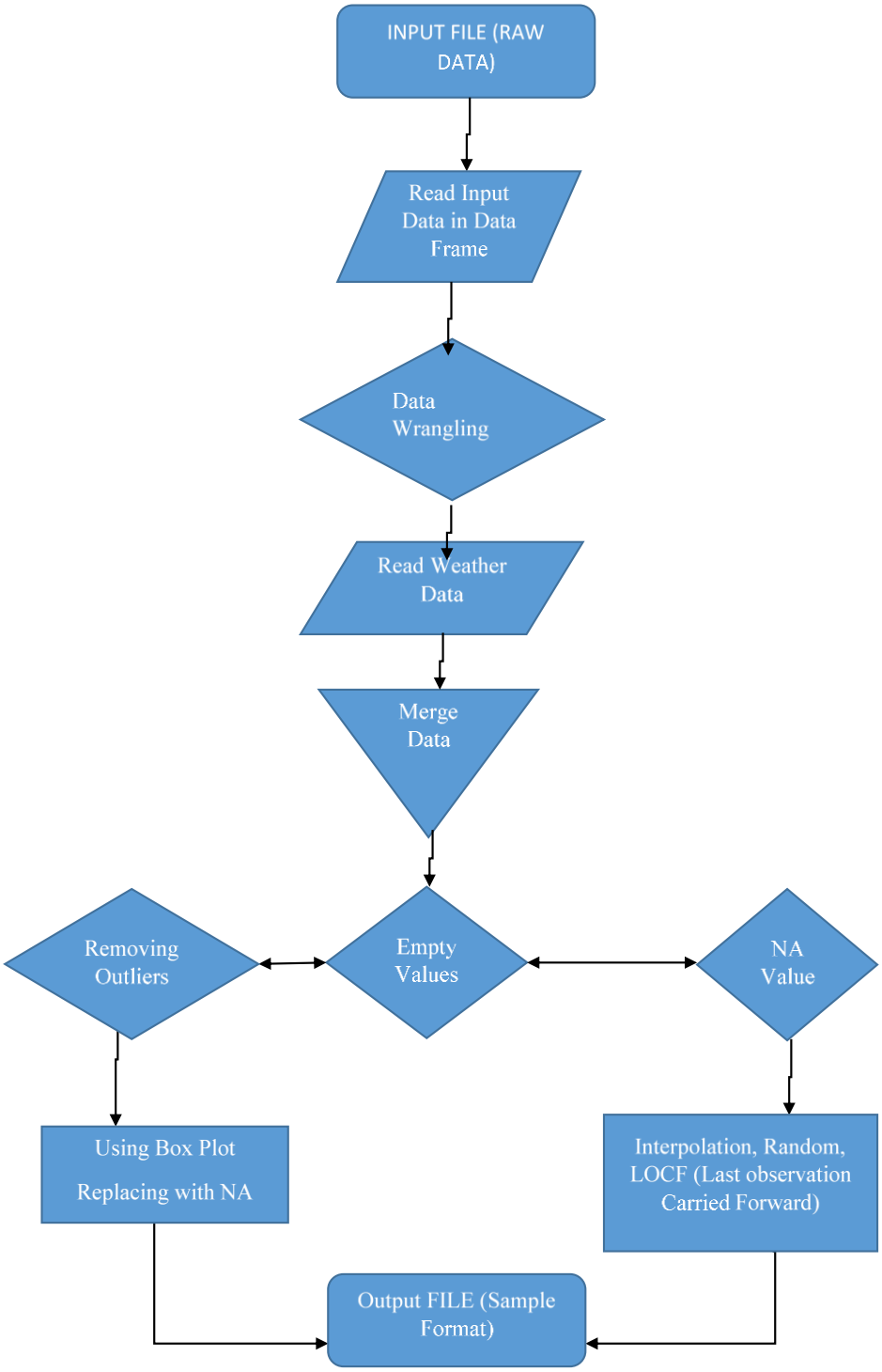
Amulya Aankul

Nand Govind Modani

Saksham Agrawal

Introduction

The data we have is of the energy consumption of a school over the span of one year. Our objective is to predict the energy use in the city of Boston using this data along with several variables from the weather data. For prediction of power consumption, we have to use Multi-linear Regression along with several other techniques and evaluate its performance. The flow chart of our project is below:



Data wrangling and Data cleaning

The steps involved in data wrangling is summarized below:-

1) Filter the unwanted and repeated rows

The first step in data wrangling was to identify what rows and columns are actually useful for us in the analysis. We noticed that the power was divided into 3 categories: Power factor, kWh and kVARh. After going through the data and other information regarding the power usage we came to a conclusion that Power factor and kVARh must be ignored as it hardly adds to the total power. Moreover, we don't get charged for the Reactive power. This leaves us with rows where channel is MILDRED SCHOOL 1. Hence we filter out the rest of the channel values.

Account	Date	Channel	Units	0:05	0:10	
26908650026	1/1/2014	507115423 1 kWh	kWh	11.13	10.98	
26908650026	1/1/2014	507115423 1 Power Factor	Power Fac	0.998931	0.99877	(
26908650026	1/1/2014	507115423 2	kVARh	0	0	
26908650026	1/1/2014	507115423 3	kWh	0	0	
26908650026	1/1/2014	507115423 4	kVARh	1.03	1.09	
26908650026	1/1/2014	MILDRED SCHOOL 1	kWh	11.13	10.98	
26908650026	1/1/2014	MILDRED SCHOOL 2	kVARh	0	0	
26908650026	1/2/2014	507115423 1 kWh	kWh	10.17	10.14	
26908650026	1/2/2014	507115423 1 Power Factor	Power Fac	0.99562	0.994115	(
26908650026	1/2/2014	507115423 2	kVARh	0	0	
26908650026	1/2/2014	507115423 3	kWh	0	0	
26908650026	1/2/2014	507115423 4	kVARh	1.91	2.21	
26908650026	1/2/2014	MILDRED SCHOOL 1	kWh	10.17	10.14	
26908650026	1/2/2014	MILDRED SCHOOL 2	kVARh	0	0	
26908650026	1/3/2014	507115423 1 kWh	kWh	10.47	10.46	
26908650026	1/3/2014	507115423 1 Power Factor	Power Fac	0.994914	0.995233	(
26908650026	1/3/2014	507115423 2	kVARh	0	0	

2) Reshaping the structure from wide format to long format

Since we had to perform calculations on the columns, it was convenient to convert the column values into rows. This made aggregation much efficient and easier.

Account	Date	Channel	Units	variable	value
26908650026	1/1/2014	MILDRED SCHOOL 1	kWh	X0.05	11.130000
26908650026	1/2/2014	MILDRED SCHOOL 1	kWh	X0.05	10.170000
26908650026	1/3/2014	MILDRED SCHOOL 1	kWh	X0.05	10.469999
26908650026	1/4/2014	MILDRED SCHOOL 1	kWh	X0.05	9.990000
26908650026	1/5/2014	MILDRED SCHOOL 1	kWh	X0.05	18.100000
26908650026	1/6/2014	MILDRED SCHOOL 1	kWh	X0.05	9.960000
26908650026	1/7/2014	MILDRED SCHOOL 1	kWh	X0.05	9.540000
26908650026	1/8/2014	MILDRED SCHOOL 1	kWh	X0.05	9.830000
26908650026	1/9/2014	MILDRED SCHOOL 1	kWh	X0.05	9.940000
26908650026	1/10/2014	MILDRED SCHOOL 1	kWh	X0.05	10.160000
26908650026	1/11/2014	MILDRED SCHOOL 1	kWh	X0.05	10.160000
26908650026	1/12/2014	MILDRED SCHOOL 1	kWh	X0.05	10.460000
26908650026	1/13/2014	MILDRED SCHOOL 1	kWh	X0.05	10.599999
26908650026	1/14/2014	MILDRED SCHOOL 1	kWh	X0.05	10.400000
26908650026	1/15/2014	MILDRED SCHOOL 1	kWh	X0.05	9.990000
26908650026	1/16/2014	MILDRED SCHOOL 1	kWh	X0.05	10.059999
26908650026	1/17/2014	MILDRED SCHOOL 1	kWh	X0.05	10.130000
26908650026	1/18/2014	MILDRED SCHOOL 1	kWh	X0.05	10.160000

3) Aggregating to hourly values:

Account	Date	hour	kWh	month	day	year	Day of Week	Weekday	Peakhour
26908650026	01/01/2014	0	132.37	1	1	2014	3	1	0
26908650026	01/01/2014	1	132.72	1	1	2014	3	1	0
26908650026	01/01/2014	2	129.03	1	1	2014	3	1	0
26908650026	01/01/2014	3	125.76	1	1	2014	3	1	0
26908650026	01/01/2014	4	129.39	1	1	2014	3	1	0
26908650026	01/01/2014	5	132.51	1	1	2014	3	1	0
26908650026	01/01/2014	6	134.93	1	1	2014	3	1	0
26908650026	01/01/2014	7	122.81	1	1	2014	3	1	1
26908650026	01/01/2014	8	119.96	1	1	2014	3	1	1
26908650026	01/01/2014	9	123.16	1	1	2014	3	1	1
26908650026	01/01/2014	10	129.11	1	1	2014	3	1	1
26908650026	01/01/2014	11	125.83	1	1	2014	3	1	1
26908650026	01/01/2014	12	120.91	1	1	2014	3	1	1
26908650026	01/01/2014	13	125.20	1	1	2014	3	1	1
26908650026	01/01/2014	14	128.59	1	1	2014	3	1	1
26908650026	01/01/2014	15	143.17	1	1	2014	3	1	1

4) Merging the forecast data and aggregated power usage data

Weather data:

Date	hour	TemperatureF	Dew_PointF	Humidity	Sea_Level_PressureIn	VisibilityMPH	Wind_SpeedMPH	Conditions	WindDirDegrees
12/30/2013	22	37.0	36.0	95	NA	2.0	2.2	Mist	290
12/31/2013	1	37.0	36.0	92	NA	2.0	2.2	Mist	290
12/31/2013	2	32.0	32.0	100	30.18	-9999.0	4.6	Overcast	350
12/31/2013	2	33.8	32.0	93	30.18	-9999.0	3.5	Overcast	330
12/31/2013	3	35.6	33.8	93	30.21	3.1	3.5	Overcast	290
12/31/2013	3	37.4	35.6	93	30.21	3.1	1.2	Overcast	310
12/31/2013	4	41.0	36.0	77	NA	2.0	Calm	Mist	NA
12/31/2013	4	39.2	37.4	93	30.21	3.1	3.5	Overcast	340
12/31/2013	4	41.0	37.4	87	30.21	5.0	2.3	Overcast	310
12/31/2013	5	41.0	39.2	93	30.18	5.0	1.2	Overcast	290
12/31/2013	5	42.8	39.2	87	30.18	5.0	1.2	Overcast	0
12/31/2013	6	42.8	39.2	87	30.15	5.0	Calm	Overcast	0
12/31/2013	6	42.8	37.4	81	30.15	6.2	1.2	Overcast	0
12/31/2013	7	44.0	36.0	65	NA	6.0	Calm	Overcast	NA
12/31/2013	7	44.6	37.4	76	30.15	6.2	3.5	Overcast	140

Aggregated Data:

Account	Date	hour	kWh	month	day	year	Day of Week	Weekday	Peakhour
26908650026	01/01/2014	0	132.37	1	1	2014	3	1	0
26908650026	01/01/2014	1	132.72	1	1	2014	3	1	0
26908650026	01/01/2014	2	129.03	1	1	2014	3	1	0
26908650026	01/01/2014	3	125.76	1	1	2014	3	1	0
26908650026	01/01/2014	4	129.39	1	1	2014	3	1	0
26908650026	01/01/2014	5	132.51	1	1	2014	3	1	0
26908650026	01/01/2014	6	134.93	1	1	2014	3	1	0
26908650026	01/01/2014	7	122.81	1	1	2014	3	1	1
26908650026	01/01/2014	8	119.96	1	1	2014	3	1	1
26908650026	01/01/2014	9	123.16	1	1	2014	3	1	1
26908650026	01/01/2014	10	129.11	1	1	2014	3	1	1
26908650026	01/01/2014	11	125.83	1	1	2014	3	1	1
26908650026	01/01/2014	12	120.91	1	1	2014	3	1	1
26908650026	01/01/2014	13	125.20	1	1	2014	3	1	1
26908650026	01/01/2014	14	128.59	1	1	2014	3	1	1

Merged Data on Date, Account and hour:

Date	hour	Account	kWh	month	day	year	Day of We	Weekday	Peakhour	Temperatu	Dew_Poin	Humidity	Sea_Level_	VisibilityM	Wind_Spe	Conditions	WindDirDegrees
1/1/2014	0	2.69E+10	132.37	1	1	2014	3	1	0	NA	NA	NA	NA	NA	NA	Mostly Clo	330
1/1/2014	1	2.69E+10	132.72	1	1	2014	3	1	0	41	30	55	NA	NA	0	Mostly Clo	330
1/1/2014	2	2.69E+10	129.03	1	1	2014	3	1	0	42.8	33.8	70	30.12	6.2	4.6	Mostly Clo	70
1/1/2014	3	2.69E+10	125.76	1	1	2014	3	1	0	44.6	33.8	66	30.12	6.2	2.3	Mostly Clo	50
1/1/2014	4	2.69E+10	129.39	1	1	2014	3	1	0	44.6	33.8	66	30.12	6.2	3.5	Mostly Clo	0
1/1/2014	5	2.69E+10	132.51	1	1	2014	3	1	0	44.6	33.8	66	30.12	6.2	5.8	Overcast	360
1/1/2014	6	2.69E+10	134.93	1	1	2014	3	1	0	46.4	33.8	62	30.09	6.2	5.8	Mostly Clo	10
1/1/2014	7	2.69E+10	122.81	1	1	2014	3	1	1	46.4	32	57	30.09	6.2	3.5	Mostly Clo	0
1/1/2014	8	2.69E+10	119.96	1	1	2014	3	1	1	46.4	33.8	62	30.12	6.2	4.6	Mostly Clo	110
1/1/2014	9	2.69E+10	123.16	1	1	2014	3	1	1	44.6	33.8	66	30.12	6.2	2.3	Mostly Clo	140
1/1/2014	10	2.69E+10	129.11	1	1	2014	3	1	1	42.8	33.8	70	30.12	6.2	4.6	Mostly Clo	80
1/1/2014	11	2.69E+10	125.83	1	1	2014	3	1	1	41	32	70	30.12	6.2	6.9	Overcast	60
1/1/2014	12	2.69E+10	120.91	1	1	2014	3	1	1	41	32	70	30.15	6.2	4.6	Overcast	60
1/1/2014	13	2.69E+10	125.2	1	1	2014	3	1	1	42.8	30.2	61	30.15	6.2	4.6	Overcast	170
1/1/2014	14	2.69E+10	128.59	1	1	2014	3	1	1	42.26667	30.17778	61.44444	30.14769	6.2	4.333333	Overcast	330
1/1/2014	15	2.69E+10	143.17	1	1	2014	3	1	1	41.73333	30.15556	61.88889	30.14538	6.2	4.066667	Overcast	330
1/1/2014	16	2.69E+10	140	1	1	2014	3	1	1	41.2	30.13333	62.33333	30.14308	6.2	3.8	Overcast	330
1/1/2014	17	2.69E+10	138.64	1	1	2014	3	1	1	40.66667	30.11111	62.77778	30.14077	6.2	3.533333	Overcast	330
1/1/2014	18	2.69E+10	143.2	1	1	2014	3	1	1	40.13333	30.08889	63.22222	30.13846	6.2	3.266667	Overcast	330

5) Removing the duplicate, outliers and NA values:

We identified the outliers from Boxplot method and removed NA values using the following logic:

- Missing values in numerical columns like TemperatureF, Dew_PointF, Humidity, Sea_Level_PressureIn, VisibilityMPH and Wind_SpeedMPH were handled by using prediction by Interpolation method.

- Missing values in Conditions column was handled by using Last observation carried forward (LOCF) method.
- Missing values in WindDirDegrees were simply filled up using random values as the values were random.

Due to the categorical values in the row, it was difficult to aggregate the duplicate key values, so we simply decided to consider the last value.

Date	hour	TemperatureF	Dew_PointF	Humidity	Sea_Level_PressureIn	VisibilityMPH	Wind_SpeedMPH	Conditions	WindDirDegrees
12/30/2013	22	37.0	36.0	95	NA	2.0	2.2	Mist	290
12/31/2013	1	37.0	36.0	92	NA	2.0	2.2	Mist	290
12/31/2013	2	32.0	32.0	100	30.18	-9999.0	4.6	Overcast	350
12/31/2013	2	33.8	32.0	93	30.18	-9999.0	3.5	Overcast	330
12/31/2013	3	35.6	33.8	93	30.21	3.1	3.5	Overcast	290
12/31/2013	3	37.4	35.6	93	30.21	3.1	1.2	Overcast	310

The clean data looks something like this, and is ready for modelling:

1	Date	hour	Account	kWh	month	day	year	Day of We	Weekday	Peakhour	Temperatu	Dew_Poin	Humidity	Sea_Level	VisibilityM	Wind_Spe	Conditions	WindDirDe
2	1/1/2014	0	2.69E+10	132.37	1	1	2014	3	1	0	NA	NA	NA	NA	NA	NA	Mostly Clo	330
3	1/1/2014	1	2.69E+10	132.72	1	1	2014	3	1	0	41	30	55	NA	NA	0	Mostly Clo	330
4	1/1/2014	2	2.69E+10	129.03	1	1	2014	3	1	0	42.8	33.8	70	30.12	6.2	4.6	Mostly Clo	70
5	1/1/2014	3	2.69E+10	125.76	1	1	2014	3	1	0	44.6	33.8	66	30.12	6.2	2.3	Mostly Clo	50
6	1/1/2014	4	2.69E+10	129.39	1	1	2014	3	1	0	44.6	33.8	66	30.12	6.2	3.5	Mostly Clo	0
7	1/1/2014	5	2.69E+10	132.51	1	1	2014	3	1	0	44.6	33.8	66	30.12	6.2	5.8	Overcast	360
8	1/1/2014	6	2.69E+10	134.93	1	1	2014	3	1	0	46.4	33.8	62	30.09	6.2	5.8	Mostly Clo	10
9	1/1/2014	7	2.69E+10	122.81	1	1	2014	3	1	1	46.4	32	57	30.09	6.2	3.5	Mostly Clo	0
10	1/1/2014	8	2.69E+10	119.96	1	1	2014	3	1	1	46.4	33.8	62	30.12	6.2	4.6	Mostly Clo	110
11	1/1/2014	9	2.69E+10	123.16	1	1	2014	3	1	1	44.6	33.8	66	30.12	6.2	2.3	Mostly Clo	140
12	1/1/2014	10	2.69E+10	129.11	1	1	2014	3	1	1	42.8	33.8	70	30.12	6.2	4.6	Mostly Clo	80
13	1/1/2014	11	2.69E+10	125.83	1	1	2014	3	1	1	41	32	70	30.12	6.2	6.9	Overcast	60
14	1/1/2014	12	2.69E+10	120.91	1	1	2014	3	1	1	41	32	70	30.15	6.2	4.6	Overcast	60
15	1/1/2014	13	2.69E+10	125.2	1	1	2014	3	1	1	42.8	30.2	61	30.15	6.2	4.6	Overcast	170
16	1/1/2014	14	2.69E+10	128.59	1	1	2014	3	1	1	42.26667	30.17778	61.44444	30.14769	6.2	4.333333	Overcast	330
17	1/1/2014	15	2.69E+10	143.17	1	1	2014	3	1	1	41.73333	30.15556	61.88889	30.14538	6.2	4.066667	Overcast	330
18	1/1/2014	16	2.69E+10	140	1	1	2014	3	1	1	41.2	30.13333	62.33333	30.14308	6.2	3.8	Overcast	330
19	1/1/2014	17	2.69E+10	138.64	1	1	2014	3	1	1	40.66667	30.11111	62.77778	30.14077	6.2	3.533333	Overcast	330
20	1/1/2014	18	2.69E+10	143.2	1	1	2014	3	1	1	40.13333	30.08889	63.22222	30.13846	6.2	3.266667	Overcast	330
21	1/1/2014	19	2.69E+10	147.12	1	1	2014	3	1	0	39.6	30.06667	63.66667	30.13615	6.2	3	Overcast	330
22	1/1/2014	20	2.69E+10	147.83	1	1	2014	3	1	0	39.06667	30.04444	64.11111	30.13385	6.2	2.733333	Overcast	330
23	1/1/2014	21	2.69E+10	134.88	1	1	2014	3	1	0	38.53333	30.02222	64.55556	30.13154	6.2	2.466667	Overcast	330
24	1/1/2014	22	2.69E+10	128.91	1	1	2014	3	1	0	38	30	65	30.12923	6.2	2.2	Overcast	340
25	1/1/2014	23	2.69E+10	124.5	1	1	2014	3	1	0	38.33333	30	63.66667	30.12692	6.2	1.466667	Overcast	330
26	1/2/2014	0	2.69E+10	121.13	1	2	2014	4	1	0	38.66667	30	62.33333	30.12462	6.2	0.733333	Overcast	330
27	1/2/2014	1	2.69E+10	125.78	1	2	2014	4	1	0	39	30	61	30.12231	6.2	0	Overcast	330
28	1/2/2014	2	2.69E+10	128.33	1	2	2014	4	1	0	37.4	33.8	87	30.12	6.2	2.3	Mostly Clo	330
29	1/2/2014	3	2.69E+10	130.99	1	2	2014	4	1	0	39.2	33.8	81	30.15	6.2	0	Mostly Clo	0

final sample format

Code:

```
setwd("C:/Users/amuly/Desktop/Fall 2016/ADS/Assignment 2")
#Reading CSV into dataframe

df1 <- read.csv("rawData1.csv")

df2 <- read.csv("rawData2.csv")
df <- rbind(df1,df2)


#Filtering the unwanted values
df_tbl <- df %>% tbl_df %>% filter (Channel == 'MILDRED SCHOOL 1')


#Reshaping the structure for calculations
mdata <- melt(as.data.frame(df_tbl), id=c("Account","Date","Channel","Units")) %>% tbl_df


#Removing "X" character for calculations
mdata$timeInterval = gsub("X","",mdata$variable)


#Dropping the variable column
mdata$variable <- NULL

#Labelling the hour
mdata$hour <- cut(as.numeric(mdata$timeInterval), seq(0,24,1),right=TRUE,label=seq(0,23,1))
```

```r
#Finding the sum per hour
aggregatedData <- mdata %>% group_by(Account, Date, hour) %>% summarise(kWh = sum(value))
View(aggregatedData)

#Splitting the Date field and converting it into Data frame
a <- strsplit(as.character(aggregatedData$Date), "/")
mat <- matrix(unlist(a), ncol=3, byrow=TRUE)
df    <- as.data.frame(mat)


#Setting the values of month,day, year and Day of the week
aggregatedData$month = df$V1
aggregatedData$day = df$V2
aggregatedData$year = df$V3

aggregatedData$"Day of Week" = wday(as.Date(aggregatedData$Date,'%m/%d/%Y')) - 1
aggregatedData$Weekday <- ifelse(aggregatedData$"Day of Week" == 0,0,
                                 ifelse(aggregatedData$"Day of Week" == 6,0,1))

#Function to calculate the Peakhour
Peakhour <- function(x){
  if(x>7 && x<=19)
    return(1)

  else
    return(0)
}
```

```r
#Applying the value to the function
aggregatedData$Peakhour <- sapply(as.numeric(aggregatedData$hour), Peakhour)

char_date <- as.character(aggregatedData$Date)
aggregatedData$Date <- format(strptime(char_date,"%m/%d/%Y"),"%m/%d/%Y")

#View the data as a table
#View(aggregatedData)


#Get distinct Date

distinctDate <- distinct(aggregatedData)

#Get latest and earliest date
a = distinctDate[order(as.Date(distinctDate$Date, format="%m/%d/%Y")),] %>% select(Date)
startDate = format(as.Date(head(a$Date,1), format="%m/%d/%Y"),"%Y-%m-%d")

endDate = format(as.Date(tail(a$Date,1), format="%m/%d/%Y"),"%Y-%m-%d")
endDate <- as.Date(endDate) +1
startDate <- as.Date(startDate) -1
```

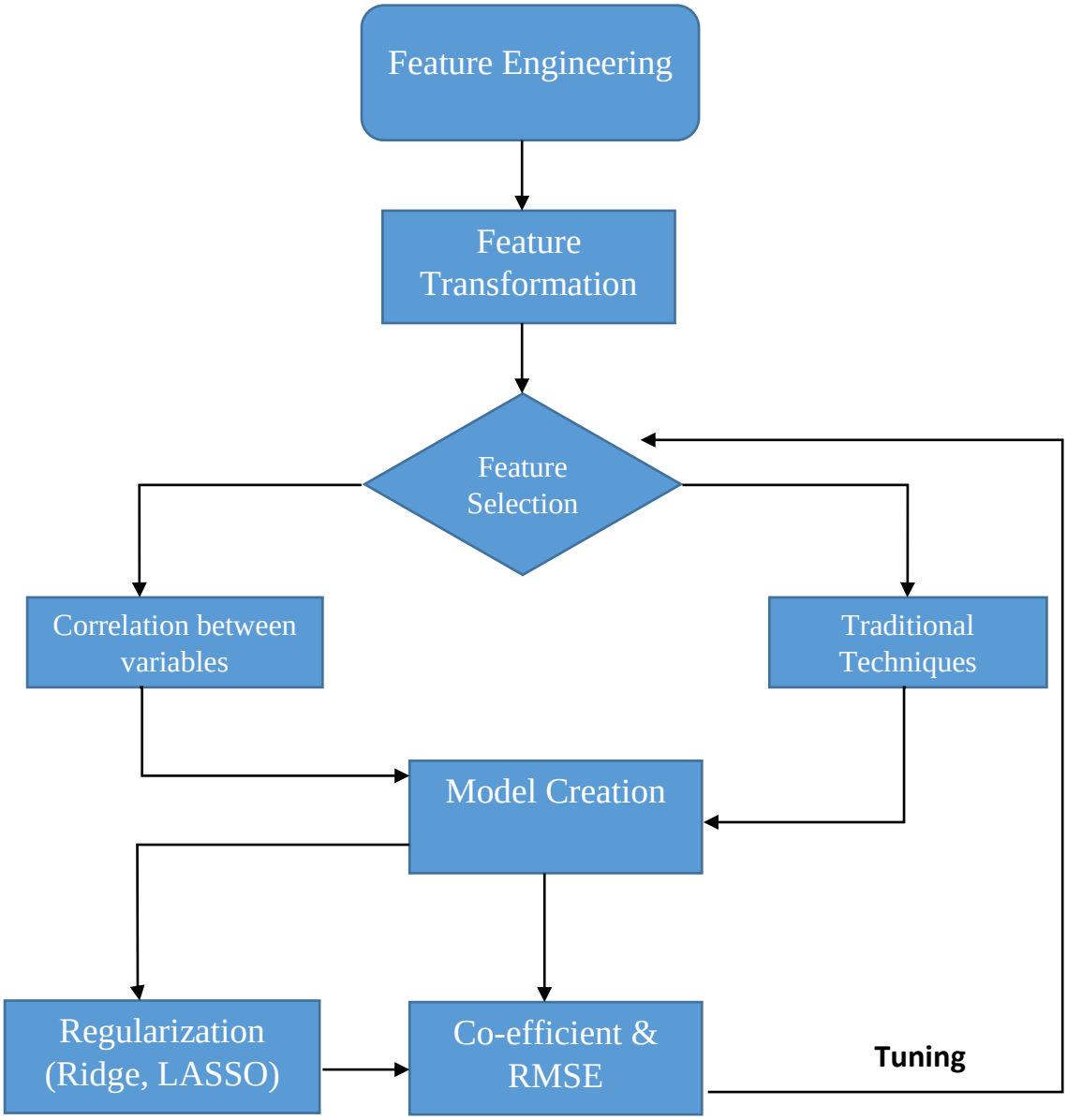
```r
#Creating a dummy DataFrame
d3 <- getWeatherForDate("BOS", start_date=startDate,
                        end_date = endDate,
                        opt_detailed = TRUE,
                        opt_all_columns = TRUE)

#Change to EST timezone
d3$Time <- as.character(d3$Time)
d3$Time <- as.POSIXct(d3$Time,tz="CET")
attributes(d3$Time)$tzone <- "America/New_York"

#Get Date
d3$Date <- format(as.Date(substr(d3$Time,1,10),format = "%Y-%m-%d"),"%m/%d/%Y")

#Get hour
d3$hour <- as.numeric(substr(d3$Time,12,13))

#Select Required Column
weatherData <- select(d3,Date,hour,TemperatureF,Dew_PointF,Humidity,Sea_Level_PressureIn,
                      VisibilityMPH,Wind_SpeedMPH,Conditions,WindDirDegrees)

View(weatherData)

#Group Data
groupedWData = sqldf("SELECT * FROM weatherData GROUP BY Date,hour")
#View(groupedWData)
```

```r
#Left outer Join
processedOp <- merge(aggregatedData, groupedWData, by=c("Date","hour"),all.x = TRUE)
#View()


#a <- merge(aggregatedData, groupedWData, by=c("Date","hour"),all.x = TRUE)
#write.csv(a,file="unclean_output_data.csv")


#Convert
processedOp$Wind_SpeedMPH[processedOp$Wind_SpeedMPH=="Calm"] <- 0

#Convert character column to numeric
processedOp$Wind_SpeedMPH <- as.double(processedOp$Wind_SpeedMPH)
#unique(processedOp$Wind_SpeedMPH)

#Remove the empty & N/A
processedOp$Conditions[processedOp$Conditions==""] <- NA
processedOp$Humidity[processedOp$Humidity=="N/A"] <- NA

processedOp$Humidity <- as.integer(processedOp$Humidity)
```

```r
#Remove the outliers using Boxplot
processedOp$TemperatureF[processedOp$TemperatureF %in% boxplot.stats(processedOp$TemperatureF)$out] <- NA
processedOp$Dew_PointF[processedOp$Dew_PointF %in% boxplot.stats(processedOp$Dew_PointF)$out] <- NA
processedOp$Humidity[processedOp$Humidity %in% boxplot.stats(processedOp$Humidity)$out] <- NA
processedOp$Sea_Level_PressureIn[processedOp$Sea_Level_PressureIn %in% boxplot.stats(processedOp$Sea_Level
processedOp$VisibilityMPH[processedOp$VisibilityMPH %in% boxplot.stats(processedOp$VisibilityMPH)$out] <-
processedOp$Wind_SpeedMPH[processedOp$Wind_SpeedMPH %in% boxplot.stats(processedOp$Wind_SpeedMPH)$out] <-

#Replacing NA values with linear interpolation
processedOp$TemperatureF <- na.approx(processedOp$TemperatureF,na.rm = FALSE)
processedOp$Dew_PointF <- na.approx(processedOp$Dew_PointF,na.rm = FALSE)
processedOp$Humidity <- na.approx(processedOp$Humidity,na.rm = FALSE)
processedOp$Sea_Level_PressureIn <- na.approx(processedOp$Sea_Level_PressureIn,na.rm = FALSE)
processedOp$VisibilityMPH <- na.approx(processedOp$VisibilityMPH,na.rm = FALSE)
processedOp$Wind_SpeedMPH <- na.approx(processedOp$Wind_SpeedMPH,na.rm = FALSE)
#Replacing NA values randomly
processedOp$WindDirDegrees[is.na(processedOp$WindDirDegrees)]<- sample(1:36,1)*10

#Replacing NA with last known non null value
processedOp$Conditions <-  na.locf(processedOp$Conditions,fromLast = TRUE,na.rm = FALSE)

#View(processedOp)
write.csv(processedOp,file="final_sample_format.csv",row.names=FALSE)
```

Multi-Linear Regression

WORKFLOW



INITIAL FEATURES	TYPE
hour	Categorical
Account	*Constant*
kWh	*Output*
month	Categorical
day	Categorical
year	*Constant*
Day.of.Week	Categorical
Weekday	Categorical
Peakhour	Categorical
TemperatureF	Numerical
Dew_PointF	Numerical
Humidity	Numerical
Sea_Level_PressureIn	Numerical
VisibilityMPH	Numerical
Wind_SpeedMPH	Numerical
Conditions	Categorical
WindDirDegrees	Numerical

Feature Transformation of Categorical Variables

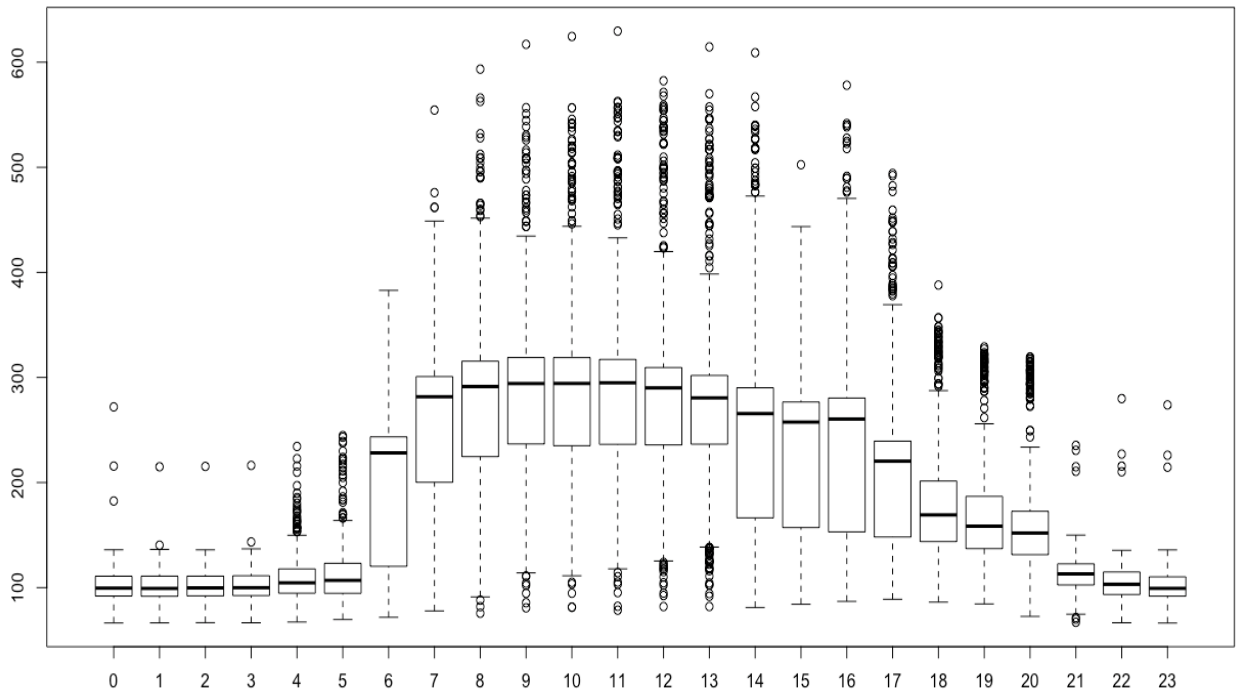
Since we have high number of levels in some categorical variables, so we have to transform them to feed it to a regression model, as it can cause over fitting.

Hour & Peak hour

Currently, hour has 24 levels of categories in it. So feeding all of them or even some of them increases the no. of variables. The box plot below shows that power consumption in between 7am to 7pm is distinctly high as compared to that of 7pm to 7am, i.e. power consumption in

Peak Hour can be easily separated from non-peak hour.

So we dropped the “hour” column and using the information from “Peak Hour” column.



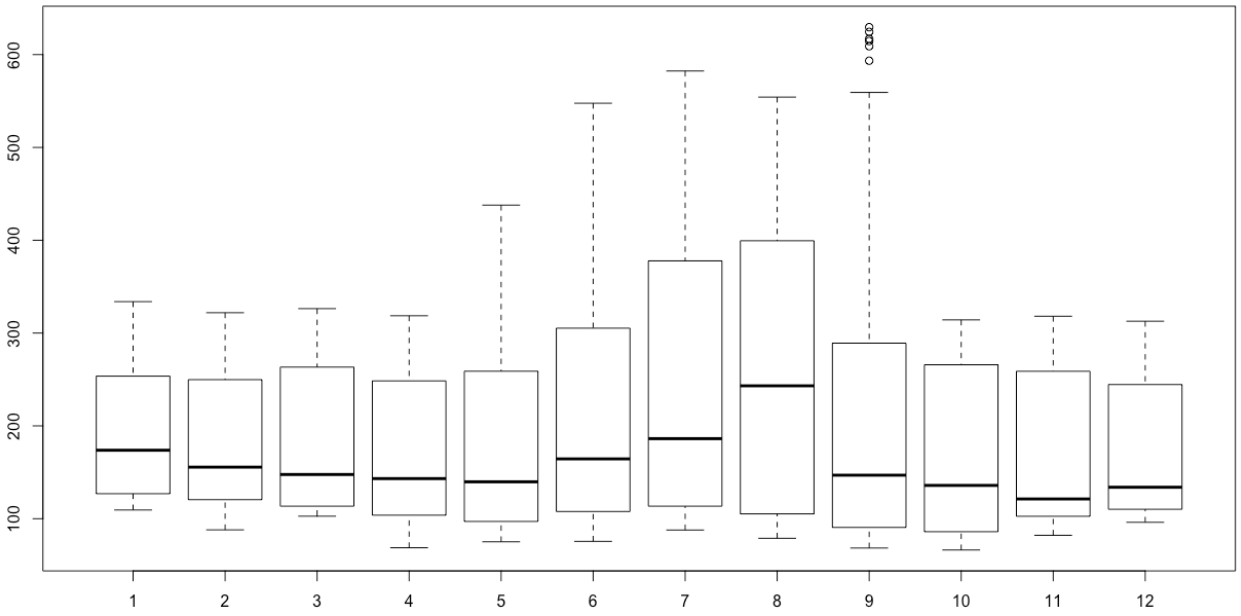
Month

“Month” has 12 levels of categories which again are high in itself to feed into any regression model. Keeping in mind that this data is of a “School” and after looking at the pattern of below plot, as similar to that of “hour”, we can reduce the categories of “month” to three, viz.

month 1 – month 4 - Spring

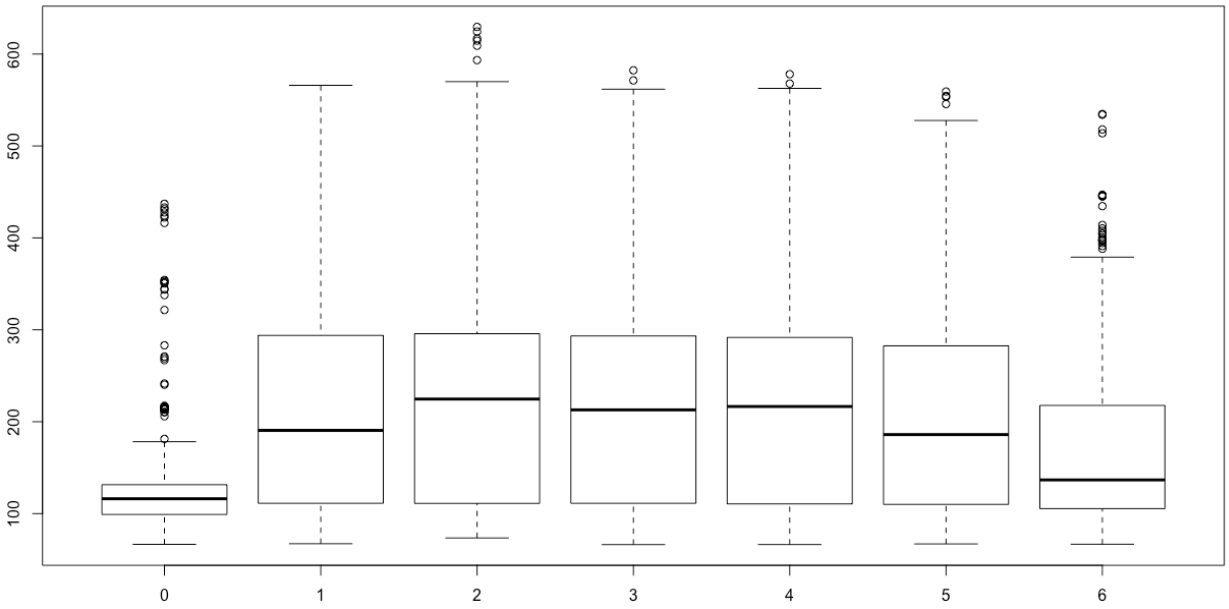
month 5 – month 8 - Summer

month 9 – month 12 - Fall

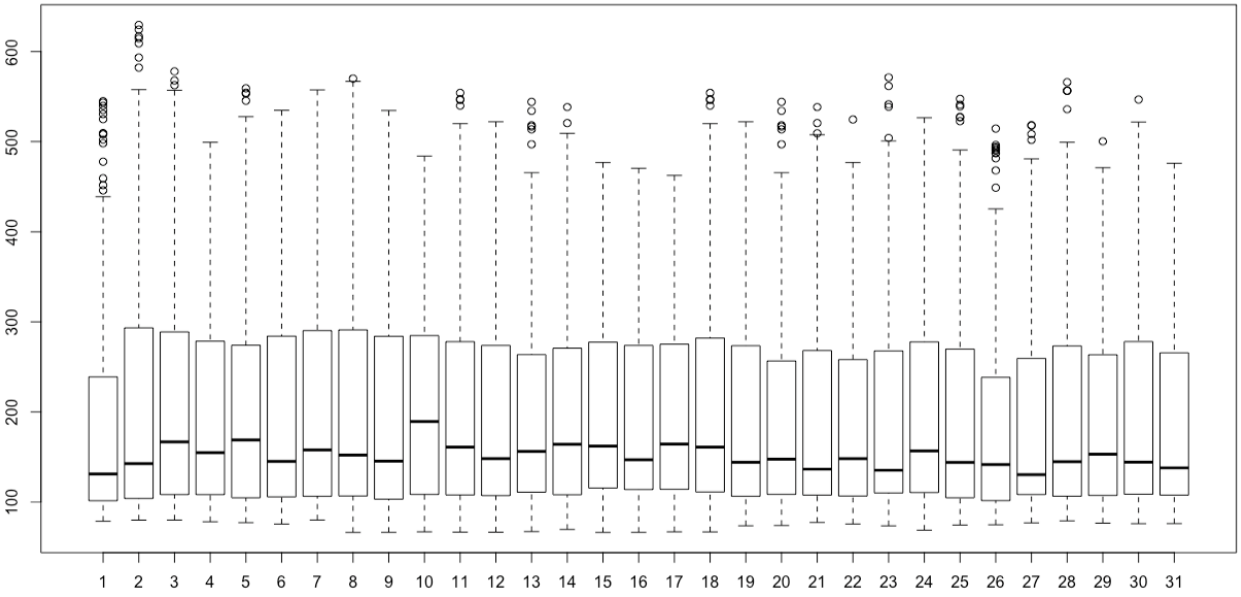


Day & Day.of.Week & Weekday

Pattern of “Day.of.Week” vs kWh clearly shows difference in levels for “Day.of.Week 0” and “Day.of.Week 6” i.e. Weekday or not. Hence, it is wise to reduce the levels of “Day.of.Week” from 7 to 2 and directly use “Weekday”.



Pattern of “Day” shows high randomness in relation to “kWh”, this is due to the fact that any “day” out of 31 levels can be a “WeekDay” for a “Month”. So instead of introducing 31 levels or even a subset of “day” categories, we can use “WeekDay” as it is more reflective of power consumption.



Conditions

Feature engineering for “Conditions” is the trickiest part as there are no fixed levels of categories. Originally, amount of missing data in “Conditions” was around 36% and around 21 categories in “Conditions” are occupying only 10% of rows. So, only 3-5 categories are present that can help in creating a prediction model. To handle this, we decided to take into account only those categories of “Conditons” which occupy more than 15% of rows and renaming all other categories as “Other”. This way we can handle any new “Conditions” value. In below screenshot, we can see the same calculation and its result in the right most column.

```
> abcd$cond <- as.character(abcd$cond)
> abcd$cond_Category <- ifelse (abcd$percent < 0.15,'other', abcd$cond)
> abcd
                          cond Freq      percent    cond_Category
1                Blowing Sand    9 0.0010291595            other
2                       Clear 1950 0.2229845626            Clear
3                     Drizzle    5 0.0005717553            other
4                         Fog   23 0.0026300743            other
5                   Heavy Fog   59 0.0067467124            other
6                  Heavy Rain    2 0.0002287021            other
7                  Heavy Snow    4 0.0004574042            other
8  Heavy Thunderstorms and Rain    3 0.0003430532            other
9                Light Drizzle   27 0.0030874786            other
10         Light Freezing Fog    8 0.0009148085            other
11                 Light Rain  445 0.0508862207            other
12                 Light Snow   26 0.0029731275            other
13         Light Thunderstorm    3 0.0003430532            other
14 Light Thunderstorms and Rain    1 0.0001143511            other
15                       Mist   23 0.0026300743            other
16              Mostly Cloudy 2113 0.2416237850    Mostly Cloudy
17                   Overcast  685 0.0783304746            other
18              Partly Cloudy 1632 0.1866209262    Partly Cloudy
19                       Rain  104 0.0118925100            other
20               Rain Showers    1 0.0001143511            other
21           Scattered Clouds 1583 0.1810177244 Scattered Clouds
22                       Snow   14 0.0016009148            other
23               Thunderstorm   13 0.0014865638            other
24      Thunderstorms and Rain    8 0.0009148085            other
25                    Unknown    4 0.0004574042            other
```

Feature Transformation of Numerical Variables

Although, we have only 7 numerical variables but we can reduce them to make a better fitting and generalized predictive model.

Temperature & Humidity

Once we fed, all numerical variables and above transformed categorical variables in Lin. Regression Model, we saw that co-efficient of "TemperatureF" is very low and hence isn't creating much of an impact. But after seeing the relation in a scatter plot, we transformed "**TemperatureF" to "SQRT(TempeatureF)**" and found a much better co-efficient.

Same Pattern was found in "Humidity", so we performed the same transformation:
"Humidity" to "SQRT(Humidity)"

```
128  fit.train <- lm(kWh~ Weekday+Peakhour+TemperatureF+Humidity,data=train)
129  summary(fit.train)
130
131:1  (Top Level)
```

```
Console ~/Documents/ADS/Assignment 2(1)/
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   13.33665    7.64776   1.744   0.0812 .
Weekday       69.72410    1.85762  37.534  < 2e-16 ***
Peakhour     126.57074    1.78371  70.959  < 2e-16 ***
TemperatureF   1.87565    0.07528  24.916  < 2e-16 ***
Humidity      -0.46337    0.06194  -7.481 8.36e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 67.8 on 6553 degrees of freedom
Multiple R-squared:  0.5978,    Adjusted R-squared:  0.5976
```

*Model Summary **Before Transformation** in Temperature & Humidity*

```
131  fit.train <- lm(kWh~ Weekday+Peakhour+sqrt(TemperatureF)+sqrt(Humidity),data=train)
132  summary(fit.train)
133
134:1  (Top Level)

Console ~/Documents/ADS/Assignment 2(1)/

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        -21.5920    14.0988  -1.531    0.126
Weekday             69.6549     1.8671  37.307   <2e-16 ***
Peakhour           126.2728     1.7941  70.383   <2e-16 ***
sqrt(TemperatureF)  24.8391     1.0840  22.914   <2e-16 ***
sqrt(Humidity)      -9.2932     0.9898  -9.389   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 68.14 on 6553 degrees of freedom
Multiple R-squared:  0.5937,    Adjusted R-squared:  0.5934
```

*Model Summary **After Transformation** in Temperature & Humidity*

FEATURE SELECTION

First step, to choose Features is to see if we have highly co-related variables.

After calculating correlation matrix for the features, we found that “TemperatureF” is highly correlated to “DewPointF”. So, this way we ruled out “DewPointF” from the model and selected “TemperatureF”

```
> correlat <- cor(xdf,method = "pearson",use="pairwise.complete.obs")
> highlyCorrelated <- findCorrelation(correlat, cutoff=0.5,verbose = TRUE,names=TRUE)
Compare row 3  and column  4 with corr  0.851
  Means:  0.22 vs 0.095 so flagging column 3
All correlations <= 0.5
> highlyCorrelated
[1] "df.TemperatureF"
```

Feeding all variables to see the model by far and try to remove variables with high p-value. So further we can see in below model that “Conditions” , “WinDirDegrees’ and “Wind_SpeedMPH” can be dropped from the model.

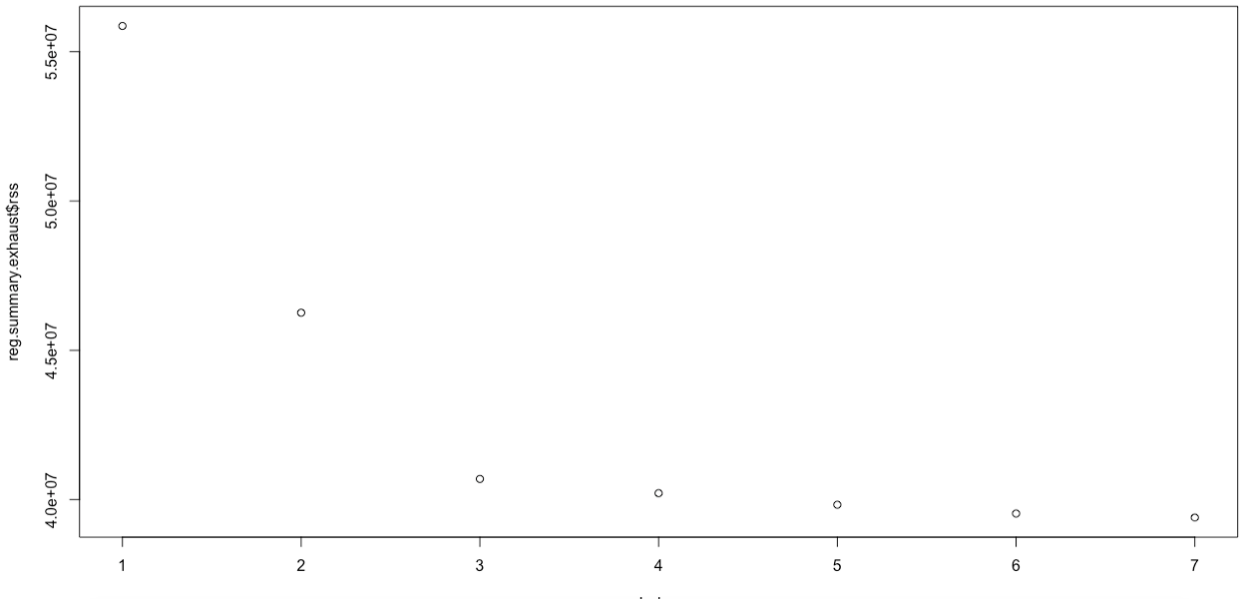
```
Call:
lm(formula = kWh ~ Sea_Level_PressureIn + Conditions + Peakhour +
    month + Weekday + Wind_SpeedMPH + WindDirDegrees + sqrt(TemperatureF) +
    sqrt(Humidity), data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-262.41  -45.56   -4.72   35.66  319.13

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                      1.104e+03  1.976e+02   5.586 2.42e-08 ***
Sea_Level_PressureIn            -3.682e+01  6.403e+00  -5.751 9.25e-09 ***
ConditionsClear                 -2.379e+01  3.929e+01  -0.605   0.5449

ConditionsScattered Clouds      -2.791e+01  3.928e+01  -0.711   0.4774
ConditionsSnow                   6.492e+00  4.678e+01   0.139   0.8896
ConditionsThunderstorm           1.032e+01  4.470e+01   0.231   0.8173
ConditionsThunderstorms and Rain -9.446e+01  4.799e+01  -1.968   0.0491 *
ConditionsUnknown                2.075e+01  5.190e+01   0.400   0.6894
Peakhour                         1.232e+02  1.839e+00  67.020  < 2e-16 ***
month                           -1.075e+00  2.685e-01  -4.004 6.29e-05 ***
Weekday                          7.120e+01  1.871e+00  38.053  < 2e-16 ***
Wind_SpeedMPH                    6.242e-01  3.139e-01   1.989   0.0468 *
WindDirDegrees                   1.771e-03  7.927e-03   0.223   0.8232
sqrt(TemperatureF)               2.619e+01  1.219e+00  21.479  < 2e-16 ***
sqrt(Humidity)                  -9.401e+00  1.132e+00  -8.306  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 67.75 on 6525 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.5957,    Adjusted R-squared:  0.5938
F-statistic: 310.1 on 31 and 6525 DF,  p-value: < 2.2e-16
```

Next Step, to create best implementation of regression, from by far deduced model. We are using **exhaustive feature selection** method to further improve the model. Below diagram shows the selectivity of each variable.

```
111  regfit.full=regsubsets (kWh~ month+Weekday+Peakhour+sqrt(Humidity)+sqrt(TemperatureF)+
112                          Sea_Level_PressureIn,data=df_test ,nvmax=7,method = "exhaustive")
113  reg.summary.exhaust =summary (regfit.full)
112:48  (Top Level)                                                          R Script
```

```
Console ~/Documents/ADS/Assignment 2(1)/
1 subsets of each size up to 7
Selection Algorithm: exhaustive
         monthSpring monthSummer Weekday Peakhour sqrt(Humidity) sqrt(TemperatureF)
1  ( 1 ) " "         " "         " "     "*"      " "             " "
2  ( 1 ) " "         " "         "*"     "*"      " "             " "
3  ( 1 ) " "         " "         "*"     "*"      " "             "*"
4  ( 1 ) " "         " "         "*"     "*"      "*"             "*"
5  ( 1 ) " "         "*"         "*"     "*"      "*"             "*"
6  ( 1 ) "*"         "*"         "*"     "*"      "*"             "*"
7  ( 1 ) "*"         "*"         "*"     "*"      "*"             "*"
         Sea_Level_PressureIn
1  ( 1 ) " "
2  ( 1 ) " "
3  ( 1 ) " "
4  ( 1 ) " "
5  ( 1 ) " "
6  ( 1 ) " "
7  ( 1 ) "*"
```

Now let's look at the RSS plot to figure out the no. of variables majorly impacting the prediction. Below graph tells us that 3 variables are enough in our model to predict the value.



MODEL

After performing feature engineering, we came to conclusion that 4 variables are enough in our model and the selected model is

kWh ~ Weekday + Peakhour+ sqrt(TemperatureF) + sqrt(Humidity)

Following is the Performance evaluation of our model:

Adjusted R-sq = 59.34 %

RMSE = 66.66879

```
Call:
lm(formula = kWh ~ Weekday + Peakhour + sqrt(TemperatureF) +
    sqrt(Humidity), data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-260.15  -44.71   -4.13   35.07  322.45

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -21.5920    14.0988  -1.531    0.126
Weekday            69.6549     1.8671  37.307   <2e-16 ***
Peakhour          126.2728     1.7941  70.383   <2e-16 ***
sqrt(TemperatureF)  24.8391     1.0840  22.914   <2e-16 ***
sqrt(Humidity)      -9.2932     0.9898  -9.389   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 68.14 on 6553 degrees of freedom
Multiple R-squared:  0.5937,    Adjusted R-squared:  0.5934
F-statistic:  2394 on 4 and 6553 DF,  p-value: < 2.2e-16
```

```
> pred = predict(fit.train, test)
> accuracy(pred, test$kWh)
               ME     RMSE      MAE       MPE     MAPE
Test set 1.83165 66.66879 50.24955 -9.512408 32.54569
>
```

REGULARIZATION

We got no improvement in RMSE after applying regularization.

Ridge Regression

We can see that RMSE got deteriorated after applying Ridge Regularization.

RMSE = 125.4789

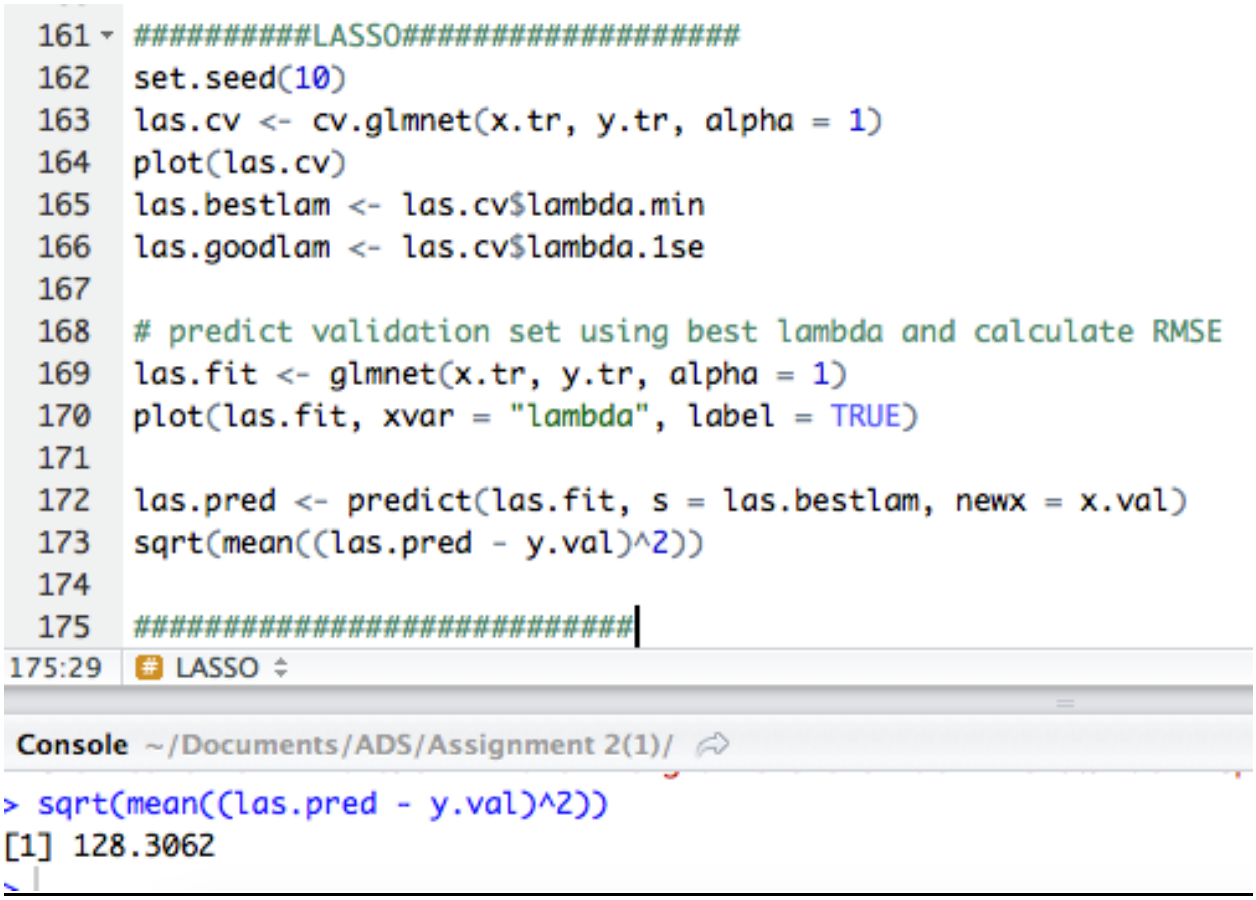
```
137
138 #######Ridge Regression##########
139 x.tr <- model.matrix(kWh ~ month+Weekday+Peakhour+sqrt(Humidity)+sqrt(TemperatureF)
140                       +Sea_Level_PressureIn, data = train)[,-1]
141 y.tr <- train$kWh
142
143 x.val <-model.matrix(kWh ~ month+Weekday+Peakhour+sqrt(Humidity)
144                       +sqrt(TemperatureF)+Sea_Level_PressureIn, data = test)[,-1]
145 y.val <- test$kWh
146
147 set.seed(10)
148 rr.cv <- cv.glmnet(x.tr, y.tr, alpha = 0)
149 plot(rr.cv)
150 rr.fit <- glmnet(x.tr, y.tr, alpha = 0)
151 rr.bestlam <- rr.cv$lambda.min
152 rr.goodlam <- rr.cv$lambda.1se
153
154 rr.fit <- glmnet(x.tr, y.tr, alpha = 0)
155 plot(rr.fit, xvar = "lambda", label = TRUE)
156
157 rr.pred <- predict(rr.fit, s = rr.bestlam, newx = x.val)
158 sqrt(mean((rr.pred - y.val)^2))
159
144:22  Ridge Regression
```

```
Console ~/Documents/ADS/Assignment 2(1)/
> sqrt(mean((rr.pred - y.val)^2))
[1] 125.4789
```

LASSO Regression

We can see that RMSE got deteriorated after applying LASSO Regularization.

RMSE = 128.3062



Forecast

Step 1 : Get data from Forcastdata.csv and store it in a data frame

Step 2 : Select the variables required for the model from Part 2

Step 3 : Get the derived variable as done in part 1

Step 4 : Run Linear Regression

Step 5 : Get the predicted value and write it into csv file

Input file:-

		Time	TimeEDT	Temperatu	Dew_Poin	Humidity	Sea_Level	VisibilityM	Wind_Dire	Wind_Spe	Gust_Spee	Precipitati	Events	Conditions	WindDirDe	DateUTC
2	1	10/1/2016 0:05	12:05 AM	57.9	57	97	30.35	10	NE	15	-	0	Rain	Light Rain	50	10/1/2016 4:05
3	2	10/1/2016 0:29	12:29 AM	57.9	57	97	30.34	10	NE	15	-	0		Overcast	50	10/1/2016 4:29
4	3	10/1/2016 0:54	12:54 AM	57.9	57	97	30.34	6	NE	18.4	-	0	Rain	Light Rain	50	10/1/2016 4:54
5	4	10/1/2016 1:16	1:16 AM	57.9	57	97	30.34	2.5	NE	19.6	-	0.01	Rain	Rain	50	10/1/2016 5:16
6	5	10/1/2016 1:23	1:23 AM	57.9	57	97	30.34	3	NE	18.4	-	0.02	Rain	Light Rain	50	10/1/2016 5:23
7	6	10/1/2016 1:54	1:54 AM	57.9	57	97	30.33	2.5	NE	16.1	-	0.08	Rain	Heavy Rair	50	10/1/2016 5:54
8	7	10/1/2016 2:03	2:03 AM	57.9	57	97	30.33	4	ENE	18.4	-	0.01	Rain	Light Rain	60	10/1/2016 6:03
9	8	10/1/2016 2:10	2:10 AM	57.9	57	97	30.33	4	NE	17.3	-	0.02	Rain	Light Rain	50	10/1/2016 6:10
10	9	10/1/2016 2:43	2:43 AM	57.9	57	97	30.32	6	NE	17.3	24.2	0.04	Rain	Light Rain	50	10/1/2016 6:43
11	10	10/1/2016 2:54	2:54 AM	57.9	57	97	30.31	5	NE	16.1	-	0.04	Rain	Light Rain	50	10/1/2016 6:54
12	11	10/1/2016 3:21	3:21 AM	57	57	100	30.31	3	NE	19.6	-	0.01	Rain	Light Rain	50	10/1/2016 7:21
13	12	10/1/2016 3:54	3:54 AM	57	57	100	30.3	4	NE	18.4	27.6	0.02	Rain	Light Rain	50	10/1/2016 7:54
14	13	10/1/2016 4:16	4:16 AM	57	57	100	30.3	6	NE	18.4	-	0.01	Rain	Light Rain	50	10/1/2016 8:16
15	14	10/1/2016 4:54	4:54 AM	57	57	100	30.3	2.5	NE	15	-	0.02	Rain	Light Rain	50	10/1/2016 8:54
16	15	10/1/2016 5:19	5:19 AM	57	57	100	30.3	4	NE	17.3	-	0.07	Rain	Light Rain	50	10/1/2016 9:19
17	16	10/1/2016 5:54	5:54 AM	57	57	100	30.31	2.5	NE	18.4	24.2	0.09	Rain	Light Rain	50	10/1/2016 9:54
18	17	10/1/2016 6:17	6:17 AM	57	57	100	30.31	3	NE	16.1	-	0.02	Rain	Light Rain	40	10/1/2016 10:17
19	18	10/1/2016 6:30	6:30 AM	57	57	100	30.31	2.5	NE	13.8	23	0.04	Rain	Light Rain	40	10/1/2016 10:30
20	19	10/1/2016 6:39	6:39 AM	57	57	100	30.31	3	NE	17.3	-	0.05	Rain	Light Rain	40	10/1/2016 10:39
21	20	10/1/2016 6:54	6:54 AM	57	57	100	30.3	4	NE	15	24.2	0.05	Rain	Light Rain	40	10/1/2016 10:54
22	21	10/1/2016 7:47	7:47 AM	55.9	55.9	100	30.31	2	NE	13.8	-	0.12	Rain	Heavy Rair	40	10/1/2016 11:47
23	22	10/1/2016 7:54	7:54 AM	55.9	55.9	100	30.31	2.5	NE	12.7	-	0.15	Rain	Heavy Rair	40	10/1/2016 11:54
24	23	10/1/2016 8:04	8:04 AM	55.9	55.9	100	30.31	4	NNE	16.1	-	0.02	Rain	Light Rain	30	10/1/2016 12:04

Code:-

```
library(ISLR)
library(forecast)

#Read the csv file
forecastData <- read.csv("forecastData.csv")

#Select the variables from the feature selection by Part 2
selectedData <- data.frame(forecastData$Time,forecastData$TemperatureF,forecastData$Humidity)

#Derive the required variables as shown in part 1
selectedData$Date <- format(as.Date(substr(selectedData$forecastData.Time,1,10),format = "%Y-%m-%d"),"%m/%d/%Y"
selectedData$hour <- as.numeric(substr(selectedData$forecastData.Time,12,13))
selectedData$"Day of Week" = wday(as.Date(x = selectedData$Date,'%m/%d/%Y')) - 1
selectedData$Weekday <- ifelse(selectedData$"Day of Week" == 0,0,ifelse(selectedData$"Day of Week" == 6,0,1))

Peakhour <- function(x){
  if(x>7 && x<=19)
    return(1)
  else
    return(0)
}

selectedData$Peakhour <- sapply(as.numeric(selectedData$hour), Peakhour)
```

```
#Add all variables to a dataframe
dffinal <-  data.frame(selectedData$Date,selectedData$hour,selectedData$Weekday,selectedData$Peakhour,selectedD


#Change the name of the variables
dffinal$Date <- dffinal$selectedData.Date
dffinal$hour <- dffinal$selectedData.hour
dffinal$Weekday <- dffinal$selectedData.Weekday
dffinal$Peakhour <- dffinal$selectedData.Peakhour
dffinal$TemperatureF <- dffinal$selectedData.forecastData.TemperatureF
dffinal$Humidity <- dffinal$selectedData.forecastData.Humidity

#Drop the variables
dffinal$selectedData.Peakhour <- NULL
dffinal$selectedData.Date <- NULL
dffinal$selectedData.hour <- NULL
dffinal$selectedData.Weekday <- NULL
dffinal$selectedData.forecastData.TemperatureF <- NULL
dffinal$selectedData.forecastData.Humidity <- NULL

#Group the rows to remove duplicates (Selects the last row)
groupedWData = sqldf("SELECT * FROM dffinal GROUP BY Date,hour")

#Read data from school
completeData <- read.csv("final_sample_format.csv")



#Convert hour as factors for model to identify
groupedWData$hour <- factor(groupedWData$hour)
completeData$hour <- factor(completeData$hour)


#Run linear Regression with the selected variables in part 2
lm.fit=lm(kWh~ (Weekday + Peakhour + sqrt(TemperatureF) + sqrt(Humidity)), data=completeData)

#Get the predicted kWh
predicted.kwh = predict(lm.fit, groupedWData)

#Add the predicted value to the dataframe
predictedDf <- data.frame(groupedWData,predicted.kwh)

write.csv(predictedDf,file="forecastOutput.csv",row.names=FALSE)
```

Output:-

	Date	hour	Weekday	Peakhour	Temperatu	Humidity	predicted.kwh
2	10/1/2016	0	0	0	57.9	97	75.70129
3	10/1/2016	1	0	0	57.9	97	75.70129
4	10/1/2016	2	0	0	57.9	97	75.70129
5	10/1/2016	3	0	0	57	100	72.92837
6	10/1/2016	4	0	0	57	100	72.92837
7	10/1/2016	5	0	0	57	100	72.92837
8	10/1/2016	6	0	0	57	100	72.92837
9	10/1/2016	7	0	0	55.9	100	71.10697
10	10/1/2016	8	0	1	55.9	100	196.2013
11	10/1/2016	9	0	1	55.9	100	196.2013
12	10/1/2016	10	0	1	57	100	198.0227
13	10/1/2016	11	0	1	57.9	97	200.7956
14	10/1/2016	12	0	1	57.9	100	199.4999
15	10/1/2016	13	0	1	57.9	100	199.4999
16	10/1/2016	14	0	1	57	100	198.0227
17	10/1/2016	15	0	1	55.9	100	196.2013
18	10/1/2016	16	0	1	55	100	194.6977
19	10/1/2016	17	0	1	55	100	194.6977
20	10/1/2016	18	0	1	55.9	97	197.497
21	10/1/2016	19	0	1	55.9	100	196.2013
22	10/1/2016	20	0	0	55	96	71.33542
23	10/1/2016	21	0	0	55	93	72.65828
24	10/1/2016	22	0	0	55	100	69.60336
25	10/1/2016	23	0	0	55	100	69.60336
26	10/2/2016	0	0	0	55	100	69.60336