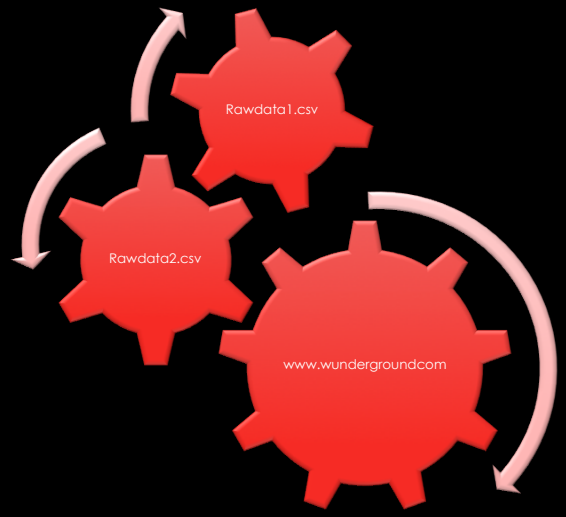
Northeastern University

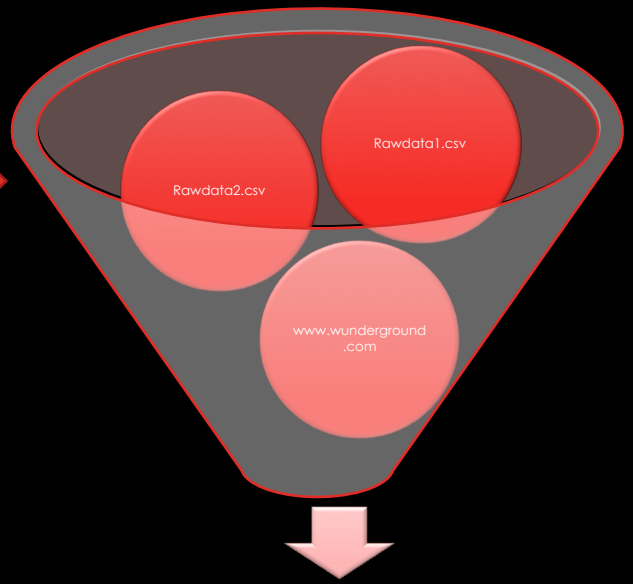
ENERGY FORCASTING

Advance Data Science - Assignment – 2

WORK FLOW



Data Wrangling & Cleaning



Feature Engineering

Linear Regression

Forward Search

Model Selection

Backward search

Stepwise (both ways)

Regularisation

Forcasting

CODE SNIPPETS

```r
install.packages("dplyr")
install.packages("tidyr")
install.packages("reshape")
library(reshape)
library(dplyr)
library(tidyr)

setwd("/Users/vigneshkarthikeyan/Desktop")
data1<- read.table("rawData1.csv",sep=",", header= TRUE)
data2<- read.table("rawData2.csv",sep=",", header= TRUE)
Result1<- data.frame(data1)
Result2<- data.frame(data2)
Result1
Result2
dplyr::tbl_df(Result1)
dplyr::tbl_df(Result2)
utils::View(Result1)
utils::View(Result2)
A <-dplyr::filter(Result1, Channel == "MILDRED SCHOOL 1")    #Filtering a specific column
utils::View(A)
#remove(B)
B<-A %>%  ungroup() %>%select(-Channel)  #Ungrouping "channel" column
utils::View(B)

y<-3
x<-y+1
y<-x+11
T_0<-(rowSums(B[,x:y]))
B<-dplyr::mutate(B,T_0)
x<-y+1
y<-x+11

T_1<-(rowSums(B[,x:y]))
B<-dplyr::mutate(B,T_1)
```

25:5 (Top Level)

- Reading data from both the csv files
- Converting data frame into tbl class
- Filtering out the Mildred school data
- Ungrouping channel column from the csv
- We combine the columns for each hour for a total of 24 hours. We use the mutate function in dplyr to merge columns with 5 min time intervals into a single column for each hour.

```
250
251 D1<-C %>% gather(hour, kWh, T_0:T_23)
252 utils::View(D1)
253
254 #Combine both csv files
255 Raw_df<-dplyr::bind_rows(D1, D)
256
257 Raw_df<-tidyr::separate(Raw_df, hour,c("n","hour"))
258 Raw_df<-Raw_df %>%  ungroup() %>%select(-n,-Units)
259 utils::View(Raw_df)
260
261 #get seperate day month year
262 Raw_df<-tidyr::separate(Raw_df, Date, c("month","day","year"))
263
264 #add Date column
265 Raw_df$Date <- as.Date(with(Raw_df, paste(year, month, day,sep="/")), "%Y/%m/%d")
266
267 #get week day from dates
268 Day_of_Week<-as.POSIXlt(Raw_df$Date)$wday
269
270 Raw_df<-dplyr::mutate(Raw_df,Day_of_Week)
271
272
273 #check for weekday
274 Raw_df$Weekday<-ifelse(Raw_df$Day_of_Week > 0 & Raw_df$Day_of_Week < 6, "1", "0")
275
276 #peek hour column
277 Raw_df <- Raw_df %>% mutate(hour = as.numeric(hour))
278 ###remove(fc_dff)
279 Raw_df$Peakhour<-ifelse(Raw_df$hour > 6 & Raw_df$hour < 20, "1", "0")
280
281 utils::View(Raw_df)
282
```

- Combine both csv files
- Separate T_0 to T and 0 to filter out hour
- Cleaning the data according to specified requirements

```r
install.packages("jsonlite")
library(jsonlite)
library(curl)
library(dplyr)
library(tidyr)
install.packages("stringr")
library("stringr")

date <-  as.Date('20140101',format = "%Y%m%d")
date2 <-  as.Date('20141231',format = "%Y%m%d")
# create a sequence of every day in this year
s <- seq(date,to = date2, by='days')

datereplace<-str_replace_all(s,"-","")

data<-paste0("http://api.wunderground.com/api/592294402e180a18/history_",datereplace[1],"/q/MA/Boston.json")
weather1<- fromJSON(data)
result<- data.frame(weather1$history$observations$date$pretty,weather1$history$observations$date$year,weather1$history$observations$date$mon,weather1$history$observat
                    weather1$history$observations$pressurei,weather1$history$observations$visi,weather1$history$observations$wspdi,
                    weather1$history$observations$wdire,weather1$history$observations$conds,weather1$history$observations$wdird)
#Result<-data.frame(lapply(Result,as.character))

for (i in 2:365)
{
  data<-paste0("http://api.wunderground.com/api/592294402e180a18/history_",datereplace[i],"/q/MA/Boston.json")
  weather1<- fromJSON(data)
  result1<- data.frame(weather1$history$observations$date$pretty,weather1$history$observations$date$year,weather1$history$observations$date$mon,weather1$history$obser
                       weather1$history$observations$pressurei,weather1$history$observations$visi,weather1$history$observations$wspdi,
                       weather1$history$observations$wdire,weather1$history$observations$conds,weather1$history$observations$wdird)
  #Result1<-data.frame(lapply(Result1,as.character))
  dplyr::tbl_df(result)
  dplyr::tbl_df(result1)
  result<-dplyr::bind_rows(result, result1)
```

- Creating a date sequence
- Traversing through all days of the year ‘2014’ and getting the required data from the api

```
}

result_1<-subset(result,weather1.history.observations.date.min == "54")
utils::View(result_1)

#To Remove extra columns
result_1<-result_1 %>%  ungroup() %>%select(-weather1.history.observations.date.pretty,-weather1.history.observations.date.min)

colnames(result_1)[1:13] = c("year","month","day","hour","Temperature","Dew_PointF","Humidity","Sea_Level_PressureIn","VisibilityMPH","Wind_SpeedMPH","Wind_Direction"
utils::View(result_1)
#Final_df<-dplyr::semi_join(Result, Raw_df,by="year","day","month","hour")#Rows that appear in either or both Result and Raw_df.
Raw_df1<- Raw_df %>% mutate(day = as.numeric(day),month = as.numeric(month),hour = as.numeric(hour))
result<- result_1 %>% mutate(day = as.numeric(day),month = as.numeric(month),hour = as.numeric(hour))
```

- To remove the extra columns
- Converting into numeric

LINEAR REGRESSION

```r
#change to factors
names <- c("year","month","day","hour","Conditions","DayofWeek","Weekday","Peakhour","WindDirDegrees")
df[,names] <- lapply(df[,names] , factor)
str(df)

#############################################################################
#discard data that have only one occurance
tab <- table(df$Conditions)
df<-df[df$Conditions %in% names(tab)[tab>2],]
#############################################################################

#75% of sample size
smp_size<-floor(0.75 * nrow(df))

#set the seed to make your partition reproductible
set.seed(123)
train_ind<-sample(seq_len(nrow(df)),size = smp_size)

#split the data into training and testing
train<-df[train_ind,]
test<-df[-train_ind,]
#fit a linear regression model

lm.fit=lm(kWh~month+Humidity+day+hour+DayofWeek+Weekday+Conditions+Peakhour+
          Temperature+Dew_PointF+Sea_Level_PressureIn+VisibilityMPH+Wind_SpeedMPH+WindDirDegrees,data=train)
summary(lm.fit)
accuracy(lm.fit)
confint(lm.fit, level=0.95)
```

OUTPUT FOR LINEAR REGRESSION

```
Console ~/Desktop/
WindDirDegrees180          -11.99061    9.24748  -1.297 0.194852
WindDirDegrees190            2.89949    8.02414   0.361 0.717866
WindDirDegrees200          -11.49784    8.58989  -1.339 0.180820
WindDirDegrees210          -15.57714    8.14451  -1.913 0.055891 .
WindDirDegrees220          -19.68241    7.79964  -2.524 0.011668 *
WindDirDegrees230           -7.92799    8.35971  -0.948 0.343021
WindDirDegrees240          -30.65067    8.93685  -3.430 0.000612 ***
WindDirDegrees250          -40.78126   10.13188  -4.025 5.83e-05 ***
WindDirDegrees260          -34.75521    9.30662  -3.734 0.000191 ***
WindDirDegrees270          -34.03014    8.53405  -3.988 6.83e-05 ***
WindDirDegrees280          -25.86345    8.24226  -3.138 0.001717 **
WindDirDegrees290          -16.68997    7.87975  -2.118 0.034246 *
WindDirDegrees300          -21.96890    7.92896  -2.771 0.005626 **
WindDirDegrees310          -24.49910    8.50081  -2.882 0.003979 **
WindDirDegrees320          -17.45757    8.60178  -2.030 0.042489 *
WindDirDegrees330           -9.00580    9.14212  -0.985 0.324656
WindDirDegrees340            4.39668    9.38902   0.468 0.639618
WindDirDegrees350            0.56126    9.04977   0.062 0.950551
WindDirDegrees360           -1.27171    8.94279  -0.142 0.886927
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 62.38 on 3140 degrees of freedom
Multiple R-squared:  0.7482,    Adjusted R-squared:  0.7389
F-statistic: 80.41 on 116 and 3140 DF,  p-value: < 2.2e-16

>
```

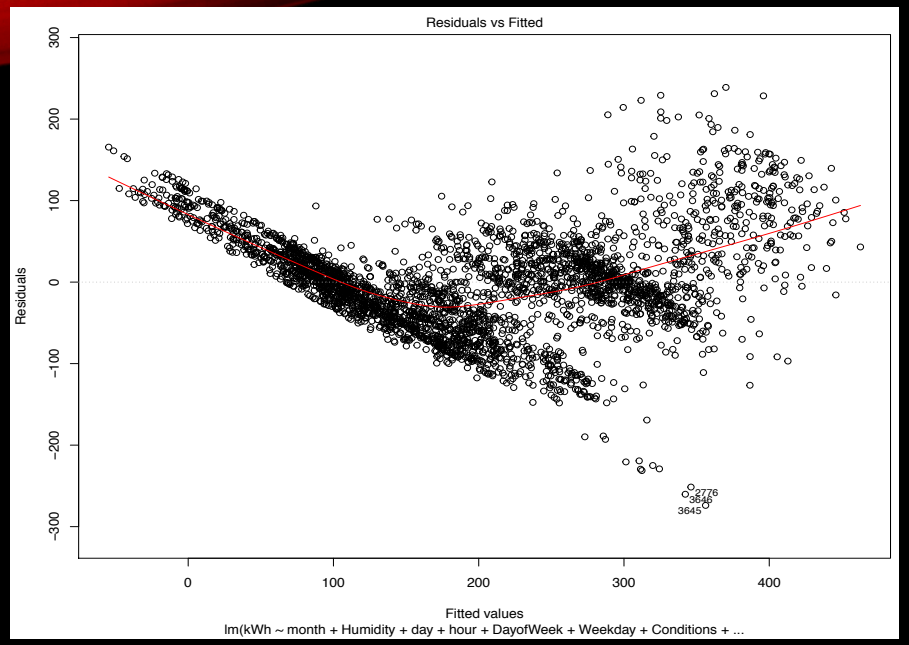
ADJUSTED R-SQUARED

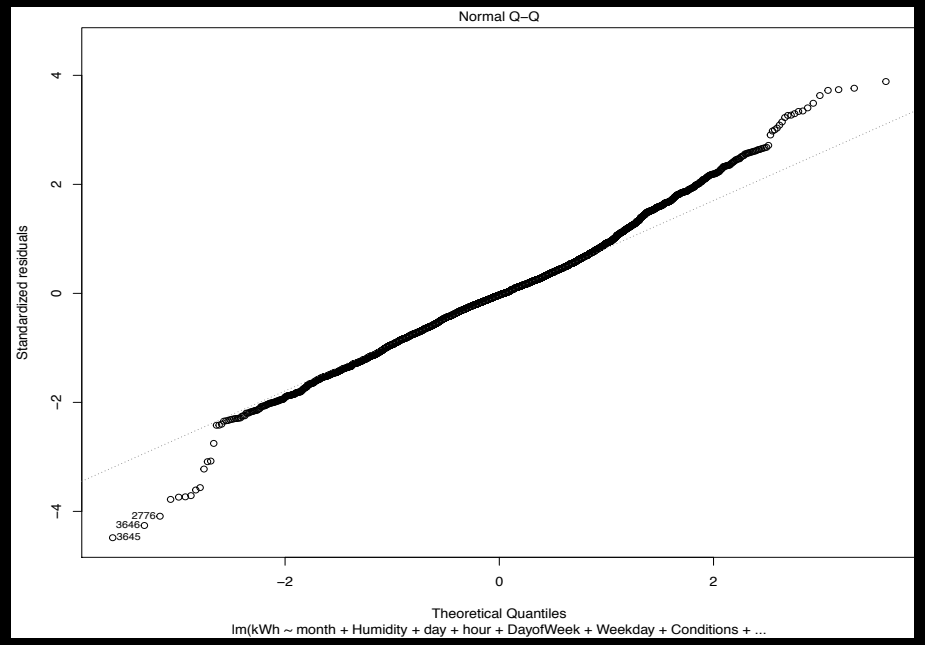
```
Console ~/Desktop/
    Sea_Level_PressureIn + VisibilityMPH + Wind_SpeedMPH + WindDirDegrees,
    data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-273.929  -38.720   -1.864   33.476  238.861

Coefficients: (2 not defined because of singularities)
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)           871.29799  231.93125   3.757 0.000175 ***
month8                 16.37206    3.95845   4.136 3.63e-05 ***
month9                -35.91381    4.18675  -8.578  < 2e-16 ***
month10               -45.54525    5.31327  -8.572  < 2e-16 ***
month11               -19.14129    7.64461  -2.504 0.012334 *
month12               -14.40230    8.08743  -1.781 0.075038 .
Humidity               -1.19586    0.52147  -2.293 0.021900 *
day2                   37.78661    8.98917   4.204 2.70e-05 ***
day3                   20.29914    9.18409   2.210 0.027160 *
day4                   -1.45858    9.00349  -0.162 0.871315
day5                   29.37147    8.86671   3.313 0.000935 ***
day6                   16.32910    8.94827   1.825 0.068121 .
day7                   23.79951    9.13138   2.606 0.009195 **
day8                   24.64827    8.72751   2.824 0.004770 **
day9                    9.43721    9.09549   1.038 0.299550
day10                  21.33571    9.00181   2.370 0.017841 *
day11                  27.29988    9.21658   2.962 0.003079 **
day12                  17.82842    9.05366   1.969 0.049019 *
```

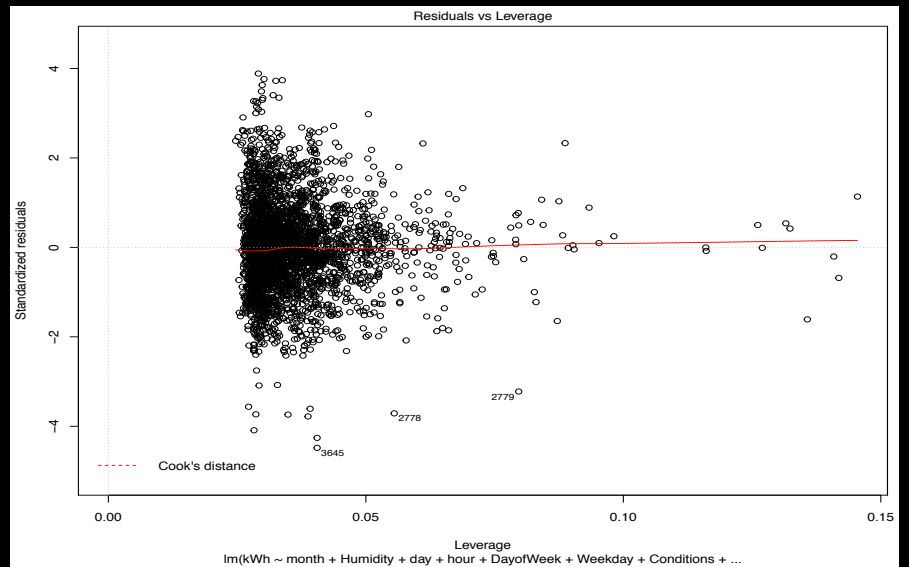
```
49  #measure of predictive accuracy
50  pred = predict(lm.fit,test)
51  accuracy(pred, train$kWh)
52
53
51:26  (Top Level)

Console ~/Desktop/

> plot(lm.fit)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
> pred = predict(lm.fit,test)
Warning message:
In predict.lm(lm.fit, test) :
  prediction from a rank-deficient fit may be misleading
> accuracy(pred, train$kWh)
                ME     RMSE      MAE       MPE     MAPE
Test set -7.909296 149.8028 119.039 -40.25121 80.30364
>
```

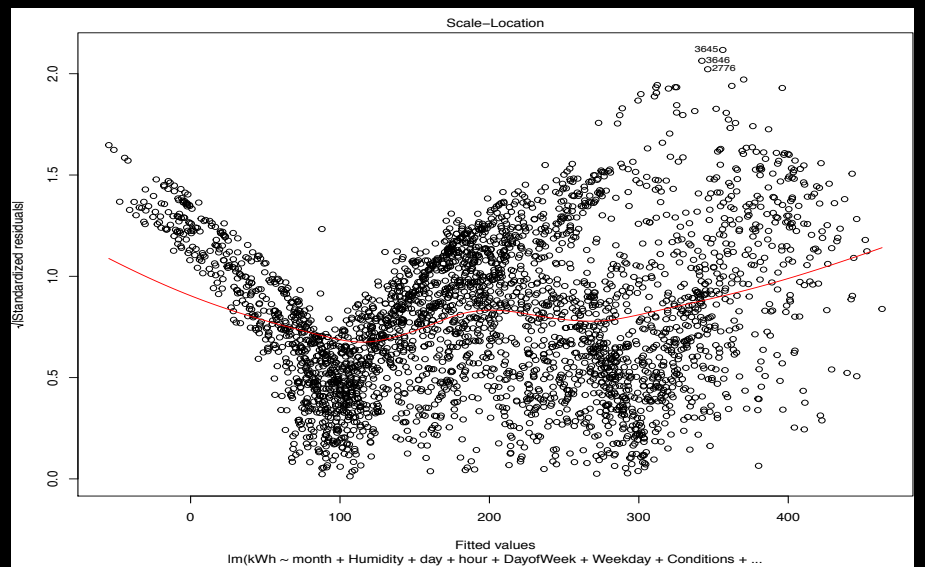
ACCURACY PREDICTION

GRAPHS


Residuals vs Fitted
Residuals
Fitted values
lm(kWh ~ month + Humidity + day + hour + DayofWeek + Weekday + Conditions + ...
2776
3646
3645


Normal Q–Q
Standardized residuals
Theoretical Quantiles
lm(kWh ~ month + Humidity + day + hour + DayofWeek + Weekday + Conditions + ...
2776
3646
3645


Residuals vs Leverage
Standardized residuals
Leverage
lm(kWh ~ month + Humidity + day + hour + DayofWeek + Weekday + Conditions + ...
Cook's distance
2779
2778
3645


Scale–Location
√|Standardized residuals|
Fitted values
lm(kWh ~ month + Humidity + day + hour + DayofWeek + Weekday + Conditions + ...
3645
3646
2776

CONFIDENCE INTERVAL FOR LINEAR REGRESSION

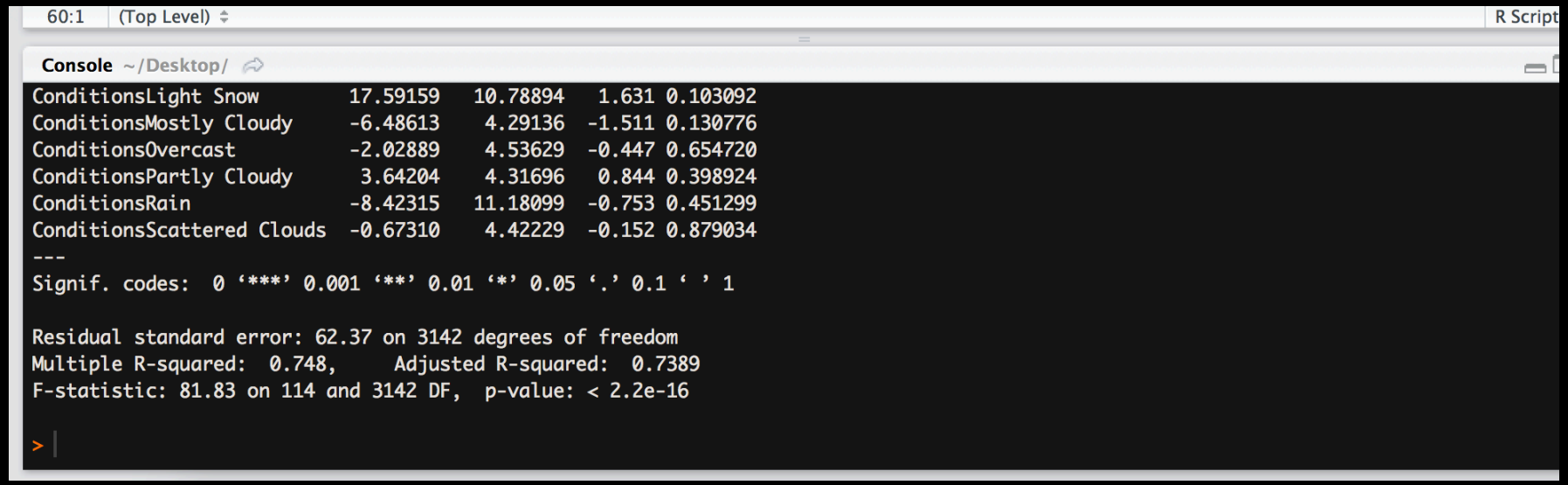
Console ~/

```
Multiple R-squared:  0.7368,    Adjusted R-squared:  0.7315
F-statistic: 139.2 on 128 and 6366 DF,  p-value: < 2.2e-16

> confint(lm.fit, level=0.95)
                                    2.5 %        97.5 %
(Intercept)                  1.321691e+02 582.292230128
month2                      -2.199247e+01  -8.416066735
month3                      -2.547092e+01 -11.638666657
month4                      -5.608990e+01 -39.819524505
month5                      -6.532940e+01 -46.384483152
month6                      -3.753913e+01 -15.412288882
month7                      -3.656580e+00  19.644737143
month8                       6.763164e+00  29.838938844
month9                      -4.687136e+01 -24.817560185
month10                     -5.874104e+01 -40.622856030
month11                     -4.277536e+01 -27.945083994
month12                     -4.109125e+01 -26.302900750
Humidity                    -1.585148e+00  -0.659794618
day2                         2.756404e+01  48.877250113
day3                         1.121721e+01  32.721893411
day4                         6.531462e+00  27.750812908
day5                         2.003393e+01  41.360754693
day6                         1.316072e+01  35.107380660
day7                         6.955750e+00  28.109300386
day8                         8.091775e+00  29.692697976
day9                         1.413624e+01  35.606425803
day10                        7.598378e+00  28.912522649
day11                        3.972276e+00  25.279127283
day12                        3.941780e+00  25.716595101
day13                        2.958797e+00  24.547476375
day14                        1.171025e+01  33.114226744
day15                        7.052559e+00  28.479909506
day16                        1.007836e+01  31.417825313
day17                        6.603612e-02  21.513727769
day18                        6.693738e+00  28.348786586
day19                        6.278115e+00  27.689912565
day20                       -2.873131e+00  18.732112249
day21                       -2.234639e+00  19.453828356
day22                       -3.062982e+00  18.562252198
```

VARIABLE SELECTION

```
55  #variable selection
56
57  #1 Backward selection in two ways
58  x_1<-step(lm(kWh~month+Humidity+day+hour+DayofWeek+Conditions+Weekday+Peakhour+Temperature+Dew_PointF+Sea_Level_PressureIn+VisibilityMPH+Wind_SpeedMPH+WindDirDegrees,
59  summary(x_1)
60  #2 Forward selection in two ways
61  x_2<-step(lm(kWh~1,data=train),direction="forward",scope=~month+Humidity+day+Conditions+hour+DayofWeek+Weekday+Peakhour+Temperature+Dew_PointF+Sea_Level_PressureIn+Vi
62  summary(x_2)
63  #regfit.fwd=regsubsets(SEC~LU+Area+Orientation+Neighbors+WoodStud+ExteriorFinish+UnfinishedAttic+FinishedRoof+RoofMaterial+UnfinishedBasement+Front+Back+Left+Right+P
64
65  #3 Stepwise method in two ways
66  fit <- lm(kWh~month+Humidity+day+DayofWeek+Conditions+hour+Weekday+Peakhour+Temperature+Dew_PointF+Sea_Level_PressureIn+VisibilityMPH+Wind_SpeedMPH+WindDirDegrees,dat
67  step <- stepAIC(fit, direction="both")
68  step$terms # display results
69  anova(lm.fit,x_1,x_2,step)
70
```

FOREWARD SEARCH



BACKWARD SELECTION

```
Console ~/Desktop/

                      Df Sum of Sq      RSS   AIC
<none>                             12224012 27036
- Conditions          10     78176 12302187 27037
- Wind_SpeedMPH        1     12571 12236583 27038
- Sea_Level_PressureIn 1     62651 12286663 27051
- Dew_PointF           1    192247 12416258 27085
- WindDirDegrees      36    567130 12791142 27112
- day                 30    593335 12817347 27131
- Humidity             1    381044 12605056 27134
- month                5   1163938 13387950 27322
- DayofWeek            6   4661614 16885625 28076
- hour                23  13404070 25628082 29401
>
```

```
Console ~/Desktop/

WindDirDegrees310          -24.89140    8.49227  -2.931 0.003402 **
WindDirDegrees320          -17.79553    8.58772  -2.072 0.038328 *
WindDirDegrees330           -9.00737    9.13754  -0.986 0.324330
WindDirDegrees340            4.24801    9.38584   0.453 0.650870
WindDirDegrees350            0.27010    9.04516   0.030 0.976180
WindDirDegrees360           -1.45990    8.93955  -0.163 0.870287
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 62.37 on 3142 degrees of freedom
Multiple R-squared:  0.7481,    Adjusted R-squared:  0.7389
F-statistic: 81.83 on 114 and 3142 DF,  p-value: < 2.2e-16
```

STEPWISE REGRESSION BOTH WAYS

```
54  #variable selection
55
56  #1 Backward selection in two ways
57  x_1<-step(lm(kWh~month+Humidity+day+hour+DayofWeek+Conditions+Weekday+Peakhour+Temperature+Dew_PointF+Sea_Level_PressureIn+Visibi
58  summary(x_1)
59  #2 Forward selection in two ways
60  x_2<-step(lm(kWh~1,data=train),direction="forward",scope=~month+Humidity+day+Conditions+hour+DayofWeek+Weekday+Peakhour+Temperatu
61  summary(x_2)
62  #regfit.fwd=regsubsets(SEC~LU+Area+Orientation+Neighbors+WoodStud+ExteriorFinish+UnfinishedAttic+FinishedRoof+RoofMaterial+Unfini
63
64  #3 Stepwise method in two ways
65  fit <- lm(kWh~month+Humidity+day+DayofWeek+Conditions+hour+Weekday+Peakhour+Temperature+Dew_PointF+Sea_Level_PressureIn+Visibilit
66  step <- stepAIC(fit, direction="both")
67  step$terms # display results
68
69  ##########################################################
70  #forecast
```

66:39 (Top Level) R Script

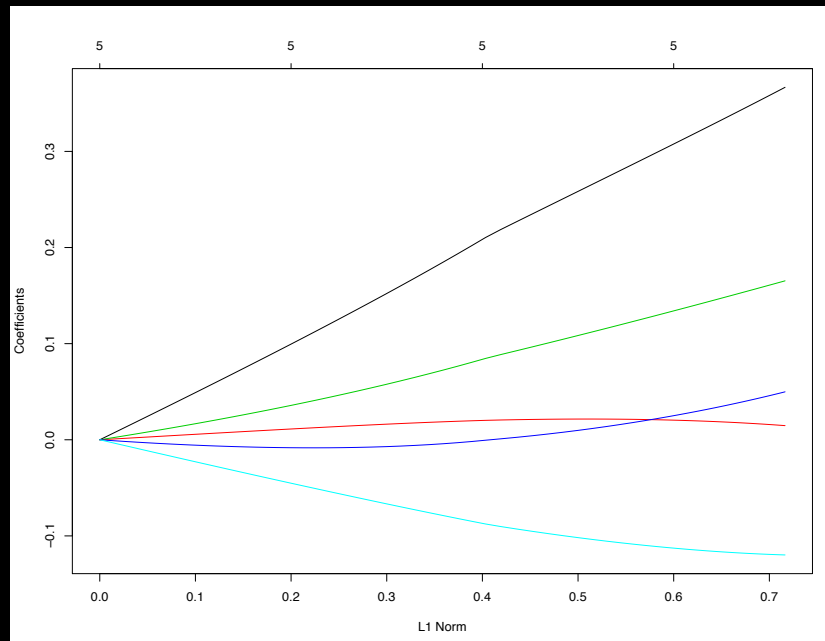
Console ~/Desktop/

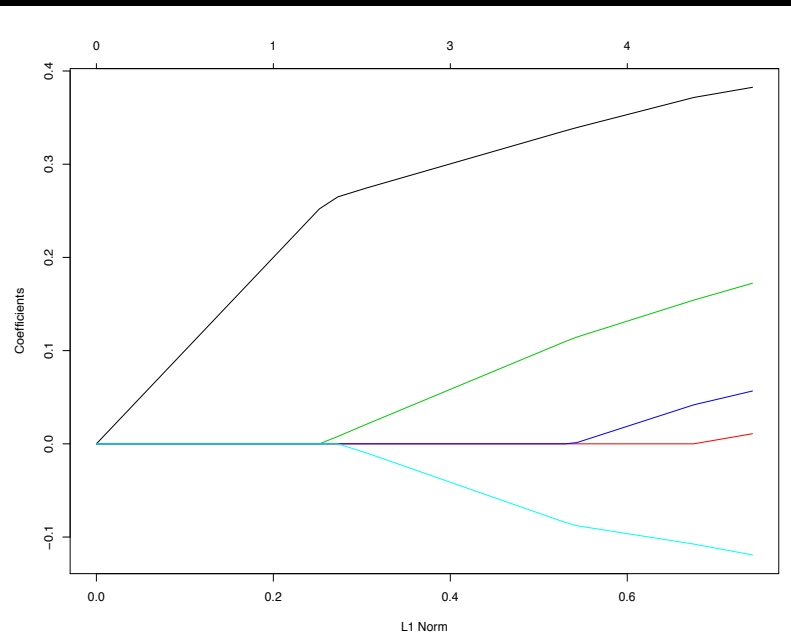
```
                    Df Sum of Sq      RSS   AIC
<none>                           12224012 27036
- Conditions        10     78176 12302187 27037
+ VisibilityMPH      1      4415 12219596 27037
- Wind_SpeedMPH     1     12571 12236583 27038
+ Temperature       1      1964 12222048 27038
- Sea_Level_PressureIn  1     62651 12286663 27051
- Dew_PointF        1    192247 12416258 27085
- WindDirDegrees   36    567130 12791142 27112
- day               30    593335 12817347 27131
- Humidity          1    381044 12605056 27134
- month             5   1163938 13387950 27322
- DayofWeek         6   4661614 16885625 28076
- hour              23  13404070 25628082 29401
```

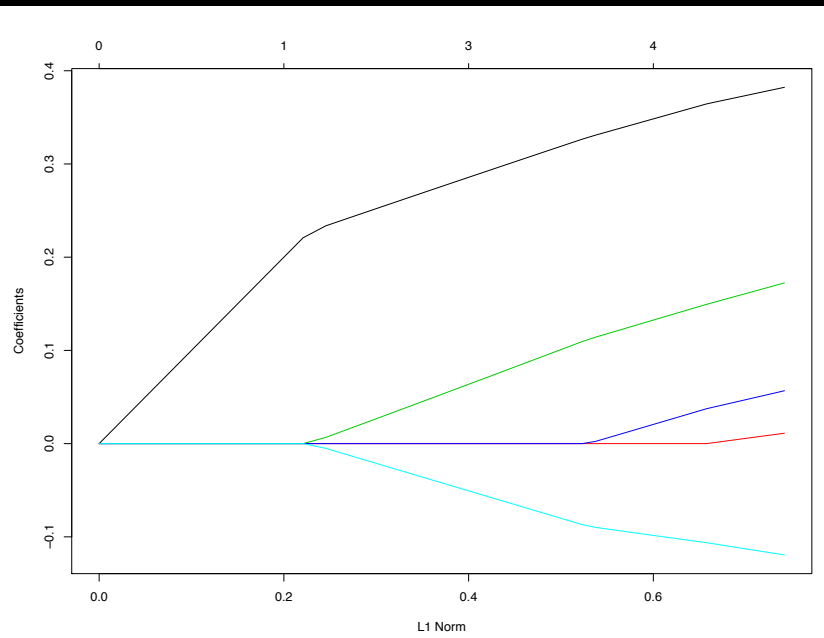
REGULARISATION

```
114  #ridge
115  install.packages("glmnet")
116  library(glmnet)
117  library("MASS")
118  install.packages("taRifx")
119  library( taRifx )
120  #dat<-train
121  dat<-transmute(train,Temperature,VisibilityMPH,Wind_SpeedMPH,kWh,Sea_Level_PressureIn,WindDirDegrees)
122  #dat <- japply( dat, which(sapply(dat, class)=="character"), as.numeric )
123  dat$WindDirDegrees<-as.numeric(dat$WindDirDegrees)
124  is.numeric(dat$WindDirDegrees)
125
126  str(dat$WindDirDegrees)
127  str(da)
128
129  class(dat$WindDirDegrees)
130
131  train_x <- as.matrix(scale(subset(dat, select = -kWh)))
132  train_y <- as.matrix(scale(dat$kWh))
133
134  cvfit = cv.glmnet(train_x, train_y)
135  plot(cvfit)
136
137  ridge <- glmnet(train_x, train_y,alpha = 0)
138  plot(ridge)
139
140  lasso <- glmnet(train_x, train_y,alpha = 1)
141  plot(lasso)
142
143  elas <- glmnet(train_x, train_y,alpha = 0.5)
144  plot(elas)
```

PLOT FOR RIDGE : LASSO : ELASTIC







Ridge Regression
$\alpha = 0$

Lasso Regression
$\alpha = 1$

Ridge Regression
$\alpha = 0.5$

FORECASTING

```
77
78  ##################################################
79  #forcast
80
81  f_df <- read.csv("forecast_data.csv",sep=",",header=TRUE)
82  attach(f_df)
83  f_df=na.omit(f_df)
84  Temperature<-fahrenheit.to.celsius(f_df$TemperatureF, round = 1)
85  f_df<-mutate(f_df,Temperature)
86  names <- c("year","month","day","hour","Conditions","DayofWeek","Weekday","Peakhour","WindDirDegrees")
87  f_df[,names] <- lapply(f_df[,names] , factor)
88  f_df[,"Wind_SpeedMPH"] <- as.numeric(as.character(f_df[,"Wind_SpeedMPH"]))
89  length(f_df$Wind_SpeedMPH)
90
91  f_df$Humidity
92
93  # from : http://www.maineharbors.com/weather/windscal.htm, Calm = 0-1 MPH
94  f_df[is.na(f_df)] <- 0
95  f_df<-f_df[complete.cases(f_df),]
96  new.kWh<-predict(lm.fit, data.frame(f_df), type="response")
97
98  fore_cast<-mutate(f_df,new.kWh)
99  write.csv(fore_cast, file = "forecastOutput.csv")
100 required<-c("Date","hour","Temperature","new.kWh")
101 fore_cast1<-transmute(fore_cast,Date,hour,Temperature,new.kWh)
102 write.csv(fore_cast1, file = "forecast_Output.csv")
103
104
105
```

FORCASTING OUTPUT

1	Day	Hr	Temp	KWH
2	5/25/2016	15	71	27.85736204
3	5/25/2016	16	73	46.46546688
4	5/25/2016	17	77	97.29683825
5	5/25/2016	18	83	50.93847402
6	5/25/2016	19	81	26.09138413
7	5/25/2016	20	79	92.65406748
8	5/25/2016	21	77	8.747102851
9	5/25/2016	22	74	97.72653271
10	5/25/2016	23	73	83.60883279
11	5/26/2016	0	72	89.45732398
12	5/26/2016	1	71	67.39894865
13	5/26/2016	2	70	32.12715716
14	5/26/2016	3	68	96.13655413
15	5/26/2016	4	67	5.224149058
16	5/26/2016	5	66	87.39167012
17	5/26/2016	6	66	8.619052359
18	5/26/2016	7	68	52.59523548
19	5/26/2016	8	71	6.856572268
20	5/26/2016	9	73	95.29163583
21	5/26/2016	10	73	28.63468249
22	5/26/2016	11	73	47.06597164
23	5/26/2016	12	75	48.51707048

forecastOutput_26435791

PERFORMANCE METRICS

```
> accuracy(x_3)
                      ME     RMSE      MAE       MPE     MAPE      MASE
Training set 8.300559e-15 68.58546 52.11594 -10.74213 33.29814 0.5814481
> accuracy(x_2)
                      ME     RMSE      MAE       MPE     MAPE      MASE
Training set 1.730565e-15 54.3612 39.17649 -5.709277 26.16338 0.436932
> accuracy(x_1)
                       ME     RMSE      MAE      MPE     MAPE      MASE
Training set -1.849993e-15 54.36688 39.17689 -5.70923 26.16453 0.4369365
> accuracy(lm.fit)
                      ME     RMSE      MAE       MPE    MAPE      MASE
Training set 3.621834e-16 54.35808 39.16446 -5.708898 26.1546 0.4367978
> 
```

Northeastern University

THANK YOU

Team 8 – Ankita Shanbhag , Sriniketan G.S. , Vignesh Karthikeyan