GSK Assignment 3

GS Kumbhare

03/08/2020

Objective

Objective is to find best classification model that can fit air quality data set. We also took relationship between error rates and model complexity. We also investigate relationship between several predictor and response variable.

Data description

The data set we have is obtained from R directory of datasets. We obtained the Airquality data for a period of 5 month. In total we have 153 instances in the dataset. In this the class is months number.

Attributes of data

1. Ozone level
2. Solar radiation
3. Windspeed
4. Temperature
5. Month
6. Day

Libraries

Libraries needed for classification model

```
library(ggthemes)
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
library(ggiraphExtra)
```

```
##
## Attaching package: 'ggiraphExtra'
```

```
## The following object is masked from 'package:ggthemes':
##
##     theme_clean
```

```
library(ggplot2)
library(broom)
```

```
library(readr)
library(MASS)
library(e1071)
library(nnet)
library(corrplot)

## corrplot 0.84 loaded

library(tidyverse)

## -- Attaching packages ---------------------------------------------------
tidyverse 1.3.0 --

## v tibble  3.0.2     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
## x dplyr::select() masks MASS::select()

library(car)

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following object is masked from 'package:purrr':
##
##     some
```

Dataset

We load our dataset in the console

```
str(airquality)

## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

From the dataset we loaded we can see that there are NA values in our data set Next we remove na values so that our model works good.

```
na<- na.omit(airquality)
str(na)
```

```
## 'data.frame':    111 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 23 19 8 16 11 14 ...
##  $ Solar.R: int  190 118 149 313 299 99 19 256 290 274 ...
##  $ Wind   : num  7.4 8 12.6 11.5 8.6 13.8 20.1 9.7 9.2 10.9 ...
##  $ Temp   : int  67 72 74 62 65 59 61 69 66 68 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 7 8 9 12 13 14 ...
##  - attr(*, "na.action")= 'omit' Named int [1:42] 5 6 10 11 25 26 27 32 33
34 ...
##   ..- attr(*, "names")= chr [1:42] "5" "6" "10" "11" ...
```

Now we summarise the clean dataset

```
summary(na)
```

```
##      Ozone          Solar.R          Wind            Temp
##  Min.   :  1.0   Min.   :  7.0   Min.   : 2.30   Min.   :57.00
##  1st Qu.: 18.0   1st Qu.:113.5   1st Qu.: 7.40   1st Qu.:71.00
##  Median : 31.0   Median :207.0   Median : 9.70   Median :79.00
##  Mean   : 42.1   Mean   :184.8   Mean   : 9.94   Mean   :77.79
##  3rd Qu.: 62.0   3rd Qu.:255.5   3rd Qu.:11.50   3rd Qu.:84.50
##  Max.   :168.0   Max.   :334.0   Max.   :20.70   Max.   :97.00
##      Month            Day
##  Min.   :5.000   Min.   : 1.00
##  1st Qu.:6.000   1st Qu.: 9.00
##  Median :7.000   Median :16.00
##  Mean   :7.216   Mean   :15.95
##  3rd Qu.:9.000   3rd Qu.:22.50
##  Max.   :9.000   Max.   :31.00
```

First we find correlation between all the variables

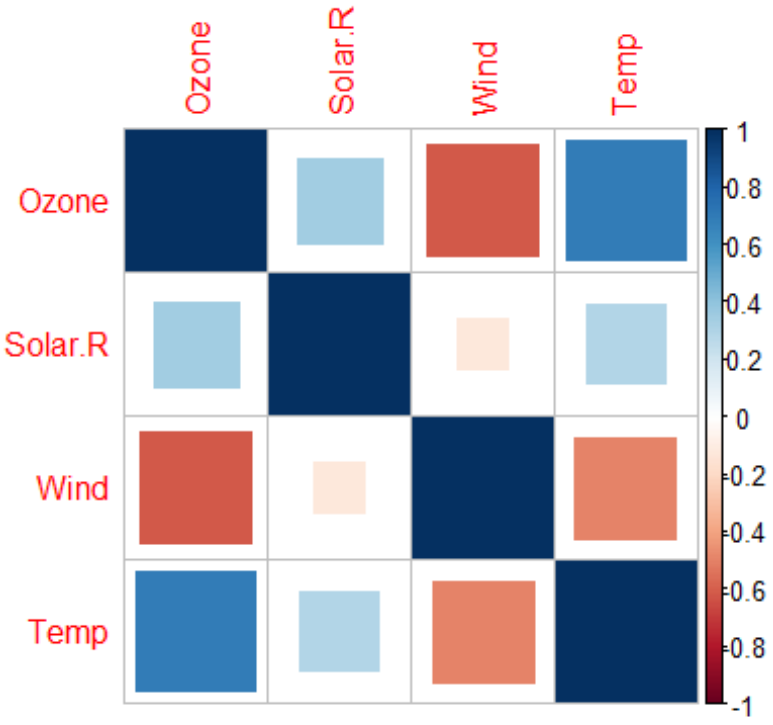
```
cor(na)
```

```
##                Ozone     Solar.R        Wind       Temp        Month
## Ozone    1.000000000  0.34834169 -0.61249658  0.6985414  0.142885168
## Solar.R  0.348341693  1.00000000 -0.12718345  0.2940876 -0.074066683
## Wind    -0.612496576 -0.12718345  1.00000000 -0.4971897 -0.194495804
## Temp     0.698541410  0.29408764 -0.49718972  1.0000000  0.403971709
## Month    0.142885168 -0.07406668 -0.19449580  0.4039717  1.000000000
## Day     -0.005189769 -0.05775380  0.04987102 -0.0965458 -0.009001079
##                 Day
## Ozone   -0.005189769
## Solar.R -0.057753801
## Wind     0.049871017
```

```
## Temp    -0.096545800
## Month   -0.009001079
## Day      1.000000000
```

We make our correlation plot for our data set

```
correlations <- cor(na[,1:4])
corrplot(correlations, method = "square")
```



Modelling

We will be using forward selection method for our modeling. In this method we will start with 1 predictor and increase to 3 predictor for each model

Our First model will be linear regression model

Linear regression Model

1. Model1 of linear regression

```
modellr1<- lm(Ozone~Solar.R, data = na)
modellr1

##
## Call:
## lm(formula = Ozone ~ Solar.R, data = na)
##
## Coefficients:
```

```
## (Intercept)      Solar.R
##     18.5987       0.1272
```

```
AIC(modellr1)
```

```
## [1] 1083.714
```

```
BIC(modellr1)
```

```
## [1] 1091.843
```

```
summary(modellr1)
```

```
##
## Call:
## lm(formula = Ozone ~ Solar.R, data = na)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.292 -21.361  -8.864  16.373 119.136
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18.59873    6.74790   2.756 0.006856 **
## Solar.R      0.12717    0.03278   3.880 0.000179 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.33 on 109 degrees of freedom
## Multiple R-squared:  0.1213, Adjusted R-squared:  0.1133
## F-statistic: 15.05 on 1 and 109 DF,  p-value: 0.0001793
```

From the above analysis of Ozone according to Solar radiation the value of residual standard error and multiple R-squared values are 31.33 and 12.13% respectively.

2. Model 2 of linear regression

```
modellr2<- lm(Ozone~Solar.R+ Wind, data = na)
modellr2
```

```
##
## Call:
## lm(formula = Ozone ~ Solar.R + Wind, data = na)
##
## Coefficients:
## (Intercept)      Solar.R         Wind
##     77.2460       0.1004      -5.4018
```

```
AIC(modellr2)
```

```
## [1] 1033.816
```

```
BIC(modellr2)
```

```
## [1] 1044.654
```

```
summary(modellr2)
```

```
##
## Call:
## lm(formula = Ozone ~ Solar.R + Wind, data = na)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.651 -18.164  -5.959  18.514  85.237
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 77.24604    9.06751   8.519 1.05e-13 ***
## Solar.R      0.10035    0.02628   3.819 0.000224 ***
## Wind        -5.40180    0.67324  -8.024 1.34e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.92 on 108 degrees of freedom
## Multiple R-squared:  0.4495, Adjusted R-squared:  0.4393
## F-statistic: 44.09 on 2 and 108 DF,  p-value: 1.003e-14
```

Our Residual standar error is 24.92 and multiple R-squared value is 44.95%.

3. Model3 of linear regression

```
modellr3<- lm(Ozone~Solar.R + Wind + Temp, data = na)
modellr3
```

```
##
## Call:
## lm(formula = Ozone ~ Solar.R + Wind + Temp, data = na)
##
## Coefficients:
## (Intercept)      Solar.R         Wind         Temp
##   -64.34208      0.05982     -3.33359      1.65209
```

```
AIC(modellr3)
```

```
## [1] 998.7171
```

```
BIC(modellr3)
```

```
## [1] 1012.265
```

```
summary(modellr3)
```

```
##
## Call:
## lm(formula = Ozone ~ Solar.R + Wind + Temp, data = na)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.485 -14.219  -3.551  10.097  95.619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -64.34208   23.05472  -2.791  0.00623 **
## Solar.R       0.05982    0.02319   2.580  0.01124 *
## Wind         -3.33359    0.65441  -5.094 1.52e-06 ***
## Temp          1.65209    0.25353   6.516 2.42e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.18 on 107 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.5948
## F-statistic: 54.83 on 3 and 107 DF,  p-value: < 2.2e-16
```

Our residual standard error and Multple R-Squared value is 21.18 and 60.59% respectively.

Summary Linear regression

With our analysis of linear regression models we see that as we increase number of predictors our value of residual standard error decreases and multiple R-Squared value increases. This shows that increase in predictor variables in reduces our error rate and increases accuracy.

Logistic regression

We will be using Logistic regression for our analysis further.

1. Model 1 of logistic regression

```
#We use logistic regression with one predictor
#1.first predictors is Solar radiation
log_fit1=glm(Ozone~Solar.R, data=na)
print(log_fit1)
```

```
##
## Call:  glm(formula = Ozone ~ Solar.R, data = na)
##
## Coefficients:
## (Intercept)      Solar.R
##     18.5987       0.1272
##
## Degrees of Freedom: 110 Total (i.e. Null);  109 Residual
## Null Deviance:       121800
## Residual Deviance: 107000    AIC: 1084
```

```
glance(log_fit1)
```

```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?

## # A tibble: 1 x 8
##   null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
##           <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
## 1       121802.     110  -539. 1084. 1092.  107022.         109   111
```

Our AIC value for model 1 of logistic regression is 1083.7

2. Model2 of logistic regression

```
#We use logistic regression with one predictor
#1.first predictors is Solar radiation
log_fit2=glm(Ozone~Solar.R + Wind, data=na)
print(log_fit2)

##
## Call:  glm(formula = Ozone ~ Solar.R + Wind, data = na)
##
## Coefficients:
## (Intercept)      Solar.R         Wind
##     77.2460       0.1004      -5.4018
##
## Degrees of Freedom: 110 Total (i.e. Null);  108 Residual
## Null Deviance:       121800
## Residual Deviance: 67050     AIC: 1034

glance(log_fit2)

## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?

## # A tibble: 1 x 8
##   null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
##           <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
## 1       121802.     110  -513. 1034. 1045.   67053.         108   111
```

In our model 2 AIC value is 1033.81. And our BIC value is 1044.65.

3. Model3 of logistic regression

```r
#We use logistic regression with one predictor
#1.first predictors is Solar radiation
log_fit3=glm(Ozone~Solar.R+ Wind + Temp, data=na)
print(log_fit3)
```

```
##
## Call:  glm(formula = Ozone ~ Solar.R + Wind + Temp, data = na)
##
## Coefficients:
## (Intercept)      Solar.R         Wind         Temp
##   -64.34208      0.05982     -3.33359      1.65209
##
## Degrees of Freedom: 110 Total (i.e. Null);  107 Residual
## Null Deviance:       121800
## Residual Deviance: 48000     AIC: 998.7
```

```r
glance(log_fit3)
```

```
## Warning: `...` is not empty.
##
## We detected these problematic arguments:
## * `needs_dots`
##
## These dots only exist to allow future extensions and should be empty.
## Did you misspecify an argument?
```

```
## # A tibble: 1 x 8
##   null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
##           <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
## 1       121802.     110  -494.  999. 1012.   48003.         107   111
```

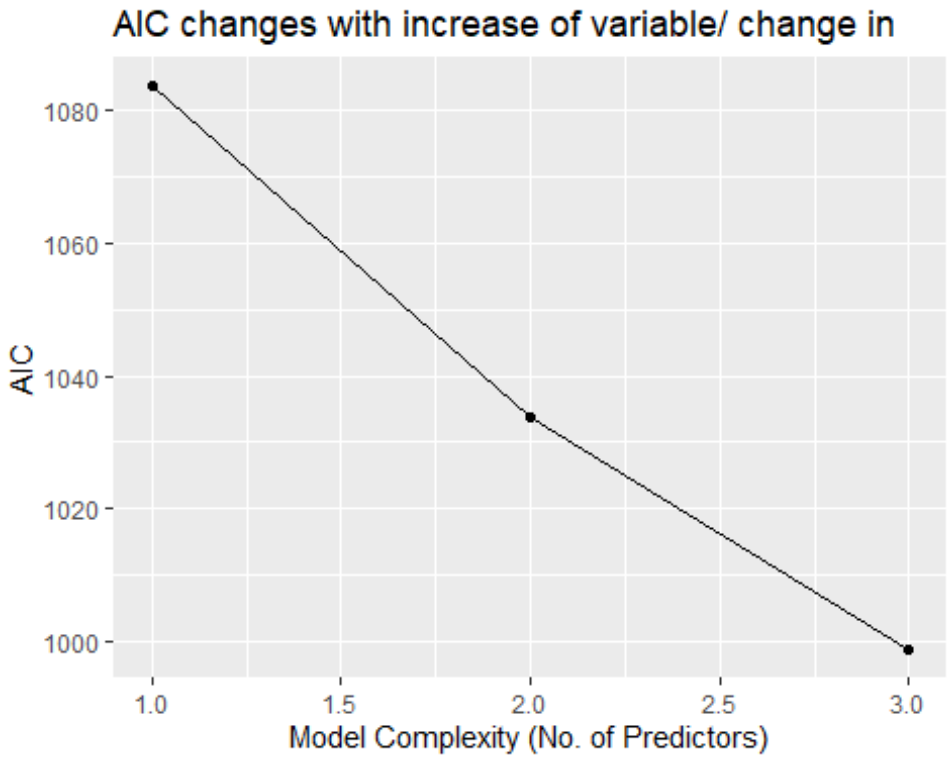
Lower AIc value tells that the model is closer to the truth. And lower BIC mean the model is considered to be true model.

Lets plot this in a graph.

```r
dat1 <- data.frame(No_of_Predictors =c(1,2,3), AIC = c(1083.7, 1033.8,
998.7171))


dat1
```

```
##   No_of_Predictors       AIC
## 1                1 1083.7000
## 2                2 1033.8000
## 3                3  998.7171
```

```r
ggplot(dat1, aes(x=No_of_Predictors, y=AIC)) +
  geom_point() +
  geom_line() +
     labs(x="Model Complexity (No. of Predictors)", y="AIC", title="AIC
changes with increase of variable/ change in ")
```



Summary of Logistic Regeression

1. We can see that as the number of variables increases the model truthfulness increases.
2. It implies that the performance of model improves as we increase the number of predictors.

KFold cross validation linear mean regression

In K-fold Cross validation the idea is to radomly divide the data into K equal sized parts.We leave out one part and fit the model to other remaining parts combined. At last we obtain prediction for teh left out part.

First we make our linear mean regression model.

1. Kfold LM

```
#kfold With linear mean regression method

set.seed(1)
train.control <- trainControl(method = "cv", number = 10)
# Train the model1
modelkfold1 <- train(Ozone ~ Solar.R , data = na , method = "lm",
                     trControl = train.control)
# Summarize the results of model 1
print(modelkfold1)

## Linear Regression
##
```

```
## 111 samples
##   1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 100, 101, 100, 98, 100, 101, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   30.63197  0.1748174  24.57872
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Our RMSE value is 30.63.

Kfold lm model 2

```
#kfold With linear mean regression method

set.seed(1)
train.control <- trainControl(method = "cv", number = 10)
# Train the model1
modelkfold2 <- train(Ozone ~ Solar.R + Wind , data = na , method = "lm",
                     trControl = train.control)
# Summarize the results of model 2
print(modelkfold2)
```

```
## Linear Regression
##
## 111 samples
##   2 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 100, 101, 100, 98, 100, 101, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   24.70793  0.4835431  20.59158
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Out RMSE value is 24.70

Kfold LM Model 3

```
#kfold With linear mean regression method

set.seed(1)
train.control <- trainControl(method = "cv", number = 10)
# Train the model3
```

```
modelkfold3 <- train(Ozone ~ Solar.R+ Wind + Temp , data = na , method =
"lm",
                     trControl = train.control)
# Summarize the results of model 3
print(modelkfold3)
```

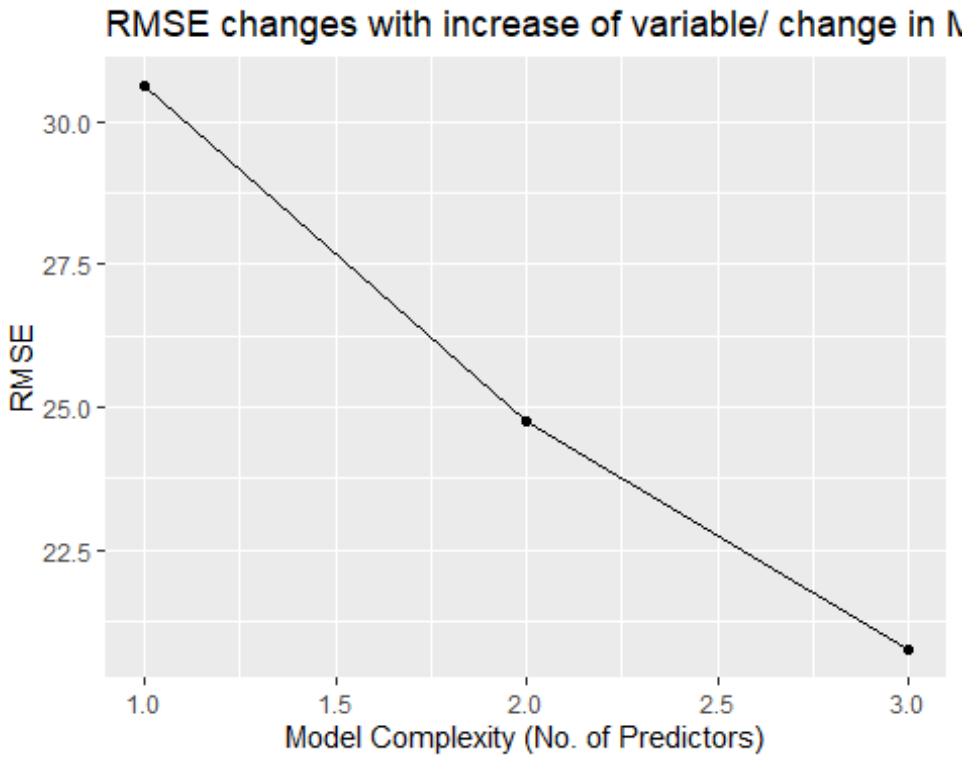
```
## Linear Regression
##
## 111 samples
##   3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 100, 101, 100, 98, 100, 101, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   20.75562  0.6568537  16.04515
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Value of RMSE has reduced to 20.75. We plot all the RMSE values.

```
dat2 <- data.frame(No_of_Predictors =c(1,2,3), RMSE= c(30.63, 24.75, 20.75))

dat2
```

```
##   No_of_Predictors  RMSE
## 1                1 30.63
## 2                2 24.75
## 3                3 20.75
```

```
ggplot(dat2, aes(x=No_of_Predictors, y=RMSE)) +
  geom_point() +
  geom_line() +
     labs(x="Model Complexity (No. of Predictors)", y="RMSE", title="RMSE
changes with increase of variable/ change in Models ")
```



Summary of Kfold LM

With lower RMSE of a model the model has better predictions. The last model with 3 predictors has lowest root mean square error. That means model with 3 prediction is better than model with lower predictors.

KNN Model

K nearest neighbour is an instance based learnign, where the function is only approximated locally and all the other computation is deferred untill function evaluation. Since this algorithm relies on distance for teh classifiation, the training dataset is normalized to increase accuracy.

1. Model1 With 1 predictor

```
#knn model 1
trControl <- trainControl(method  = "cv",
                          number  = 3)
Modelknn1 <- train(Ozone ~ Solar.R,
             method     = "knn",
             tuneGrid   = expand.grid(k = 10),
             trControl  = trControl,
             data       = na)
Modelknn1
```

```
## k-Nearest Neighbors
##
## 111 samples
##   1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 74, 74, 74
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   31.86325  0.1611655  25.75232
##
## Tuning parameter 'k' was held constant at a value of 10
```

Root mean square error value for model 1 of KNN is 30.289.

2. Model 2

```
#knn model 1
trControl <- trainControl(method  = "cv",
                          number  = 3)
Modelknn2 <- train(Ozone ~ Solar.R + Wind,
             method     = "knn",
             tuneGrid   = expand.grid(k = 10),
             trControl  = trControl,
             data       = na)
Modelknn2
```

```
## k-Nearest Neighbors
##
## 111 samples
##   2 predictor
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 74, 74, 74
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   30.54655  0.1708377  23.12547
##
## Tuning parameter 'k' was held constant at a value of 10
```

The RMSE value of model 2 of knn is 28.57. It seems like with increase in predictors the value of RMSE is drecreasing.

3. Model 3

```
#knn model 1
trControl <- trainControl(method  = "cv",
```

```
                           number  = 3)
Modelknn3 <- train(Ozone ~ Solar.R + Wind + Temp,
             method     = "knn",
             tuneGrid   = expand.grid(k = 10),
             trControl  = trControl,
             data       = na)
Modelknn3
```

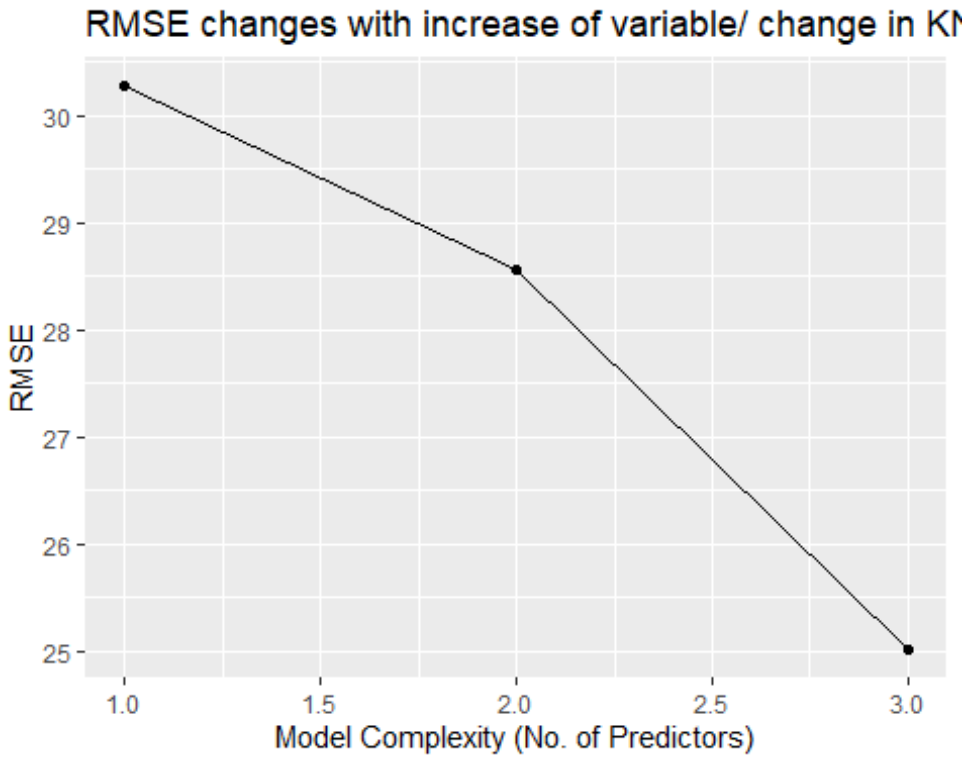
```
## k-Nearest Neighbors
##
## 111 samples
##   3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 73, 74, 75
## Resampling results:
##
##   RMSE    Rsquared  MAE
##   25.728  0.413587  19.90318
##
## Tuning parameter 'k' was held constant at a value of 10
```

Our model 3 RMSE value is 25.010.

```
dat3 <- data.frame(No_of_Predictors =c(1,2,3), RMSE= c(30.28, 28.57, 25.010))

dat3
```

```
##   No_of_Predictors  RMSE
## 1                1 30.28
## 2                2 28.57
## 3                3 25.01
```

```
ggplot(dat3, aes(x=No_of_Predictors, y=RMSE)) +
  geom_point() +
  geom_line() +
     labs(x="Model Complexity (No. of Predictors)", y="RMSE", title="RMSE
changes with increase of variable/ change in KNN Models ")
```



LOOCV Models

In LOOCV model we leave one data point and build model on the rest of dataset. Then we test the model against the data point that was left out in step one and record the test error associated with it.

Model1 LOOCV

```
#LOOCV for one predictor
train.control.loocv <- trainControl(method = "LOOCV")
# Train the model
modelloocv1 <- train(Ozone ~Solar.R, data = na, method = "lm",
               trControl = train.control.loocv)
# Summarize the results
print(modelloocv1)
```

```
## Linear Regression
##
## 111 samples
##   1 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 110, 110, 110, 110, 110, 110, ...
## Resampling results:
##
##   RMSE      Rsquared    MAE
```

```
##   31.51877  0.09617832  24.59652
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

In LOOCV model 1 our RMSE value is 31.51877 with one predictor.

2. Model 2 KNN

```
#LOOCV for one predictor
train.control.loocv <- trainControl(method = "LOOCV")
# Train the model
modelloocv2 <- train(Ozone ~Solar.R +Wind, data = na, method = "lm",
               trControl = train.control.loocv)
# Summarize the results
print(modelloocv2)
```

```
## Linear Regression
##
## 111 samples
##   2 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 110, 110, 110, 110, 110, 110, ...
## Resampling results:
##
##   RMSE     Rsquared  MAE
##   25.3448  0.415619  20.64459
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

In our second model RMSE is 25.34.

3. Model 3 LOOCV

```
#LOOCV for one predictor
train.control.loocv <- trainControl(method = "LOOCV")
# Train the model
modelloocv3 <- train(Ozone ~Solar.R + Wind + Temp, data = na, method = "lm",
               trControl = train.control.loocv)
# Summarize the results
print(modelloocv3)
```

```
## Linear Regression
##
## 111 samples
##   3 predictor
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 110, 110, 110, 110, 110, 110, ...
## Resampling results:
```

```
##
##   RMSE      Rsquared   MAE
##   21.65222  0.5734888  16.06211
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

With our 3rd model our RMSE drastically reduces to 21.6.

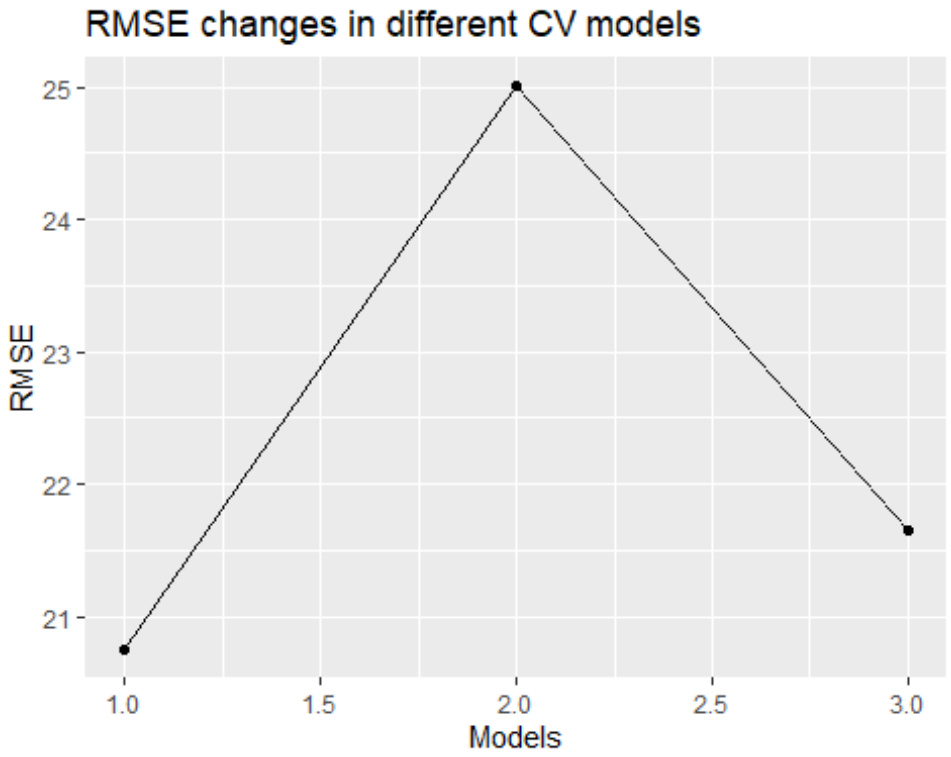
Summary

With all 3 models from LOOCV model 3 has lowest RMSE value. It shows that with increase in predictors the error rate reduces.

Conclusion

We conclude this experiment by analysis of RMSE from all the CV models we built.

```
Dat5 <- data.frame(models = c("Kfold", "KNN", "LOOCV"), RMSE= c(20.75,
25.010, 21.655))
Dat5

##   models   RMSE
## 1  Kfold 20.750
## 2    KNN 25.010
## 3  LOOCV 21.655

dat4 <- data.frame(Models =c(1,2,3), RMSE= c(20.75, 25.010, 21.655))

dat4

##   Models   RMSE
## 1      1 20.750
## 2      2 25.010
## 3      3 21.655

ggplot(dat4, aes(x=Models, y=RMSE)) +
  geom_point() +
  geom_line() +
     labs(x="Models", y="RMSE", title="RMSE changes in different CV models")
```



From all the CV models we built we see that K-fold analysis has lowest Error rate among all the CV models. Hence K-Fold is best Model for our dataset.