# R Notebook

**Name: Gabriel Bentley**

**Date: 10/03/22**

**Dataset: Gas Turbine Metric Measurements**

**https://archive.ics.uci.edu/ml/datasets/Gas+Turbine+CO+and+NOx+Emission+Data+Set**

## Data set information

The dataset contains 14795 instances of 11 sensor measures aggregated over one hour (by means of average or sum) from a gas turbine located in Turkey's north western region for the purpose of studying flue gas emissions, namely CO and NOx (NO + NO2). The data comes from the same power plant as the dataset ([Web Link]) used for predicting hourly net energy yield. By contrast, this data is collected in another data range (01.01.2011 - 31.12.2011) and (01.01.2015 - 31.12.2015), includes gas turbine parameters (such as Turbine Inlet Temperature and Compressor Discharge pressure) in addition to the ambient variables.

## Attribute information

The explanations of sensor measurements and their brief statistics are given below.

Variable (Abbr.) Unit Min Max Mean Ambient temperature (AT) C â€“6.23 37.10 17.71 Ambient pressure (AP) mbar 985.85 1036.56 1013.07 Ambient humidity (AH) (%) 24.08 100.20 77.87 Air filter difference pressure (AFDP) mbar 2.09 7.61 3.93 Gas turbine exhaust pressure (GTEP) mbar 17.70 40.72 25.56 Turbine inlet temperature (TIT) C 1000.85 1100.89 1081.43 Turbine after temperature (TAT) C 511.04 550.61 546.16 Compressor discharge pressure (CDP) mbar 9.85 15.16 12.06 Turbine energy yield (TEY) MWH 100.02 179.50 133.51 Carbon monoxide (CO) mg/m3 0.00 44.10 2.37 Nitrogen oxides (NOx) mg/m3 25.90 119.91 65.29

## Read in and seperate the data set into train and test

```
df <- read.csv("gt_2011.csv")

df$year <- rep(c("2011"), each = 7411)

colSums(is.na(df))
```

```
##   AT   AP   AH AFDP GTEP  TIT  TAT  TEY  CDP   CO  NOX year
##    0    0    0    0    0    0    0    0    0    0    0    0
```

```
df2 <- read.csv("gt_2015.csv")
df2$year <- rep(c("2015"), each = 7384)

colSums(is.na(df2))
```

```
##   AT   AP   AH AFDP GTEP  TIT  TAT  TEY  CDP   CO  NOX year
##    0    0    0    0    0    0    0    0    0    0    0    0
```

```
df3 <- rbind(df, df2)

df3$year <- factor(df3$year)
```

```
set.seed(4545)
i <- sample(1:nrow(df3), nrow(df3)*0.80, replace=FALSE)
train <- df3[i,]
test <- df3[-i,]
```

```
str(train)
```

```
## 'data.frame':    11836 obs. of  12 variables:
##  $ AT  : num  25.52 11.13 4.11 13.93 24.21 ...
##  $ AP  : num  1008 1020 1024 1009 1011 ...
##  $ AH  : num  66.1 89.2 82.5 89.7 52.3 ...
##  $ AFDP: num  3.95 4.04 3.3 3.54 3.73 ...
##  $ GTEP: num  25.7 23.9 22.6 24.1 27.2 ...
##  $ TIT : num  1093 1084 1072 1086 1093 ...
##  $ TAT : num  550 550 550 550 550 ...
##  $ TEY : num  135 133 129 134 140 ...
##  $ CDP : num  12.1 11.8 11.6 12 12.5 ...
##  $ CO  : num  0.424 1.459 3.558 1.013 1.449 ...
##  $ NOX : num  68.4 74.8 68.4 69.3 58.2 ...
##  $ year: Factor w/ 2 levels "2011","2015": 1 1 2 1 2 1 2 1 1 1 ...
```

```
summary(train)
```

```
##        AT                AP                AH               AFDP
##  Min.   :-6.235   Min.   : 989.4   Min.   : 29.27   Min.   :2.369
##  1st Qu.:10.951   1st Qu.:1009.7   1st Qu.: 64.22   1st Qu.:3.295
##  Median :16.939   Median :1013.8   Median : 76.10   Median :3.854
##  Mean   :17.133   Mean   :1014.4   Mean   : 74.01   Mean   :3.847
##  3rd Qu.:23.468   3rd Qu.:1018.3   3rd Qu.: 85.09   3rd Qu.:4.327
##  Max.   :35.822   Max.   :1036.6   Max.   :100.17   Max.   :7.319
##       GTEP            TIT             TAT              TEY
##  Min.   :17.70   Min.   :1001   Min.   :512.5   Min.   :100.0
##  1st Qu.:23.24   1st Qu.:1073   1st Qu.:543.0   1st Qu.:127.8
##  Median :25.02   Median :1086   Median :549.8   Median :133.8
##  Mean   :25.91   Mean   :1082   Mean   :545.5   Mean   :134.9
##  3rd Qu.:30.00   3rd Qu.:1100   3rd Qu.:550.0   3rd Qu.:147.5
##  Max.   :40.72   Max.   :1101   Max.   :550.6   Max.   :179.5
##       CDP               CO                NOX           year
##  Min.   : 9.871   Min.   : 0.00039   Min.   : 25.91   2011:5912
##  1st Qu.:11.545   1st Qu.: 1.09533   1st Qu.: 55.27   2015:5924
##  Median :11.980   Median : 1.78550   Median : 61.74
##  Mean   :12.156   Mean   : 2.34876   Mean   : 63.72
##  3rd Qu.:13.181   3rd Qu.: 2.97308   3rd Qu.: 70.20
##  Max.   :15.159   Max.   :43.62200   Max.   :119.32
```

```
head(train)
```

```
##            AT     AP     AH   AFDP   GTEP    TIT    TAT    TEY    CDP      CO
## 3832  25.5210 1007.5 66.106 3.9542 25.709 1092.7 550.30 134.71 12.147 0.42369
## 6989  11.1270 1019.8 89.172 4.0374 23.879 1084.1 550.00 132.84 11.797 1.45930
## 14684  4.1111 1023.7 82.495 3.3023 22.644 1072.5 549.77 129.02 11.611 3.55830
## 2650  13.9260 1009.4 89.672 3.5398 24.131 1086.2 549.96 134.26 12.018 1.01300
## 9987  24.2130 1011.1 52.348 3.7335 27.230 1092.8 550.09 139.85 12.539 1.44890
## 7163  16.2630 1005.1 94.187 4.3478 24.545 1086.8 549.95 133.75 11.895 1.62750
##          NOX year
## 3832  68.395 2011
```

```
## 6989  74.769 2011
## 14684 68.411 2015
## 2650  69.274 2011
## 9987  58.187 2015
## 7163  70.307 2011
```

```
names(train)
```

```
##  [1] "AT"   "AP"   "AH"   "AFDP" "GTEP" "TIT"  "TAT"  "TEY"  "CDP"  "CO"
## [11] "NOX"  "year"
```

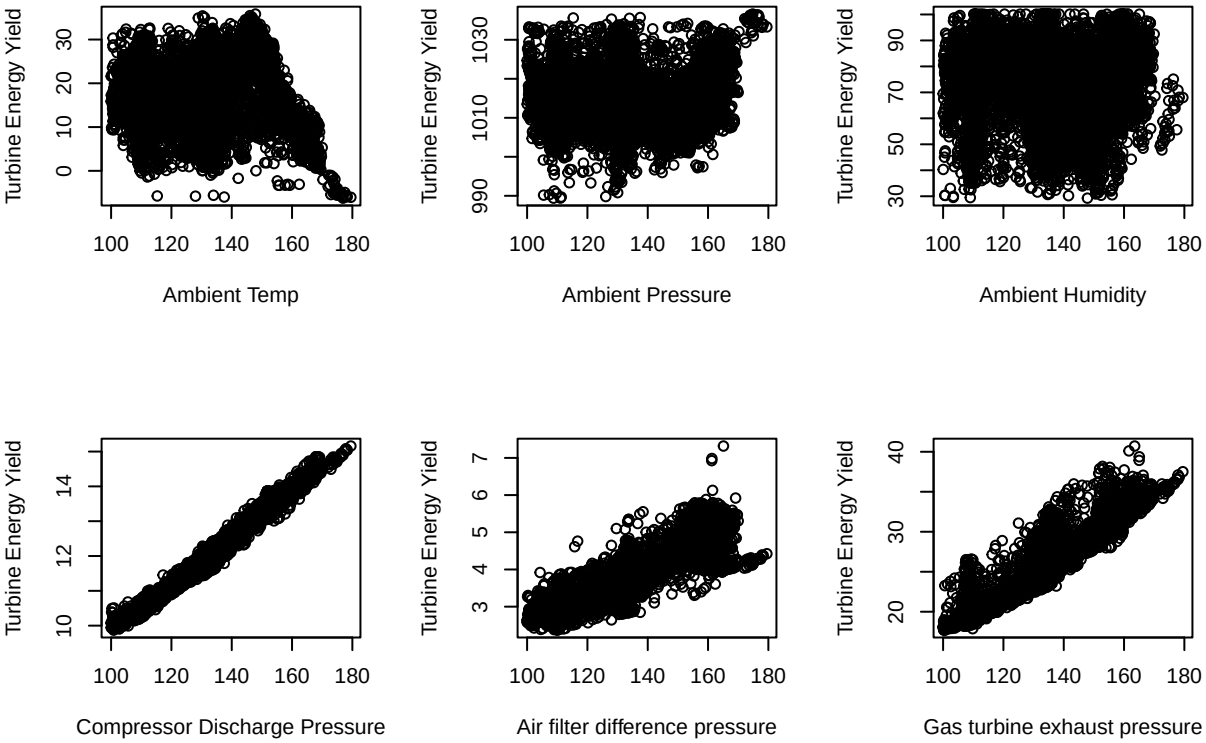
```
colSums(is.na(train))
```

```
##   AT   AP   AH AFDP GTEP  TIT  TAT  TEY  CDP   CO  NOX year
##    0    0    0    0    0    0    0    0    0    0    0    0
```

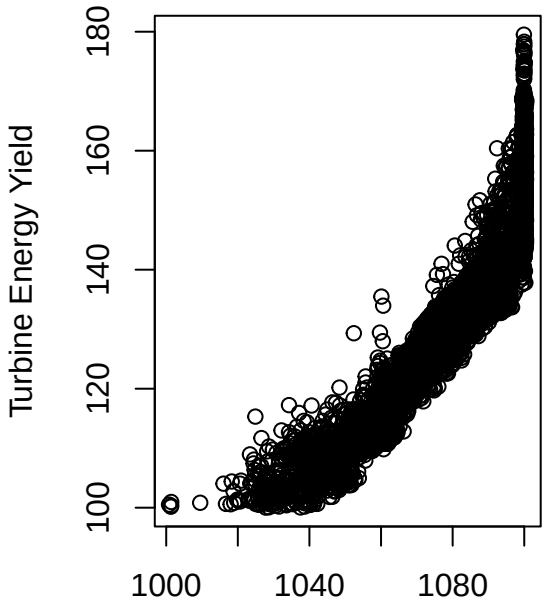
## Graphical exploration

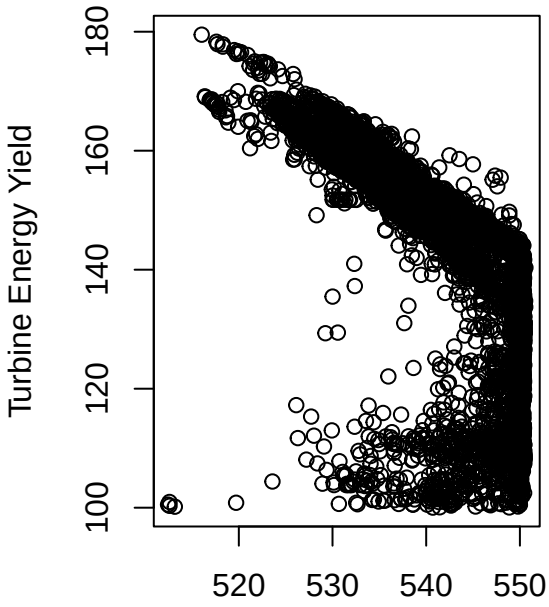
Here we plot the Turbine Energy Yield against the other predictors of the data set to determine their relation.

```
par(mfrow=c(2,3))
plot(train$TEY, train$AT, xlab="Ambient Temp", ylab="Turbine Energy Yield")
plot(train$TEY, train$AP, xlab="Ambient Pressure", ylab="Turbine Energy Yield")
plot(train$TEY, train$AH, xlab="Ambient Humidity", ylab="Turbine Energy Yield")
plot(train$TEY, train$CDP, xlab="Compressor Discharge Pressure", ylab="Turbine Energy Yield")
plot(train$TEY, train$AFDP, xlab="Air filter difference pressure", ylab="Turbine Energy Yield")
plot(train$TEY, train$GTEP, xlab="Gas turbine exhaust pressure", ylab="Turbine Energy Yield")
```

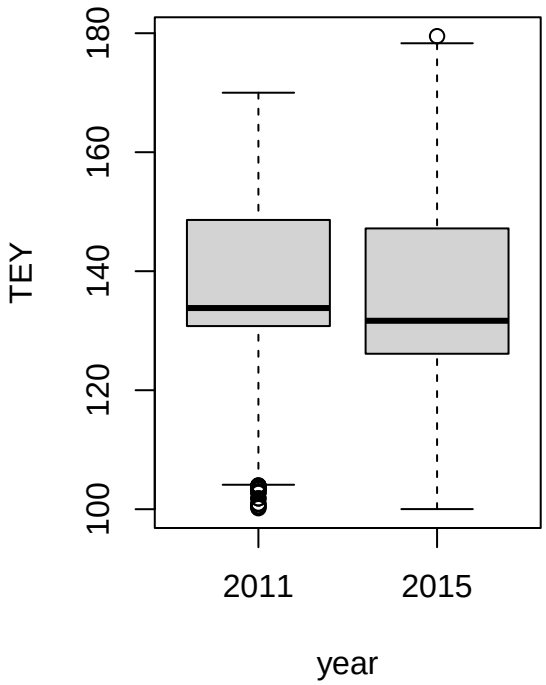



```
par(mfrow=c(1,2))
plot(train$TEY~train$TIT, xlab="Turbine Inlet Temp", ylab="Turbine Energy Yield")
plot(train$TEY~train$TAT, xlab="Turbine After Temp", ylab="Turbine Energy Yield")
```




```
plot(train$year, train$TEY, xlab="year", ylab="TEY")
```



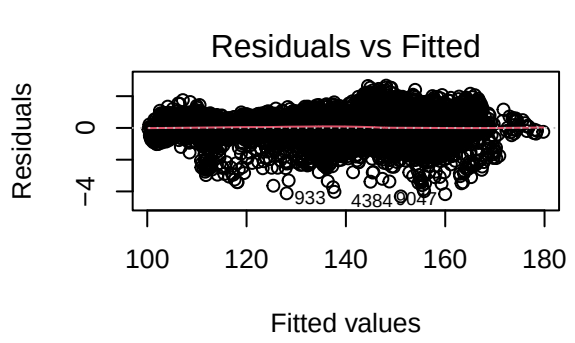

These plots tell us that the ambient data has little to do with the Turbines Energy yield, but the pressure and temperature of the turbine seems to have a linear connection with the turbines energy yield.

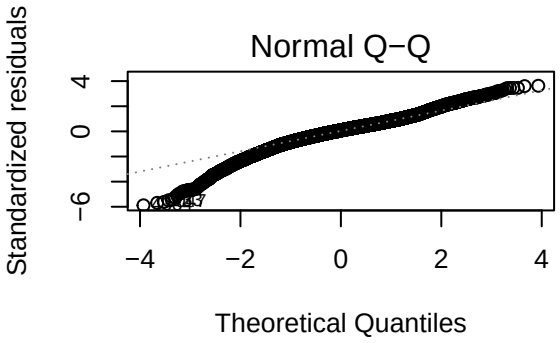
### Linear Regression model

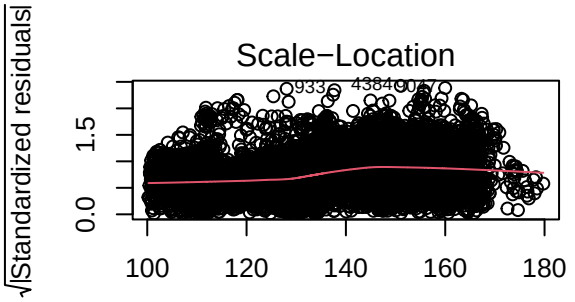
```
lm1 <- lm(TEY~., data=train)
summary(lm1)
```

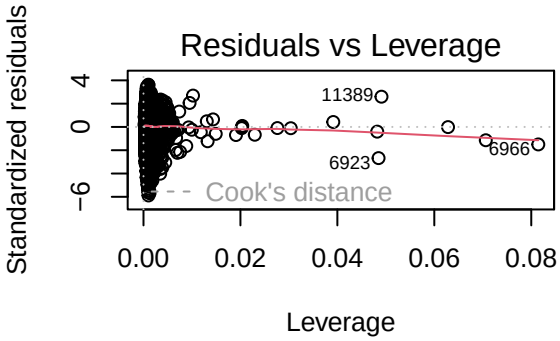
```
##
## Call:
## lm(formula = TEY ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3157 -0.3746  0.0603  0.4250  2.6619
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -1.446e+02  1.826e+00  -79.167   <2e-16 ***
## AT          -3.264e-01  1.721e-03 -189.649   <2e-16 ***
## AP          -6.851e-02  1.300e-03  -52.685   <2e-16 ***
## AH          -1.213e-02  6.924e-04  -17.514   <2e-16 ***
## AFDP        -4.294e-02  2.701e-02   -1.590   0.1119
## GTEP        -1.314e-02  6.607e-03   -1.988   0.0468 *
## TIT          6.666e-01  5.822e-03  114.490   <2e-16 ***
## TAT         -7.007e-01  8.579e-03  -81.672   <2e-16 ***
## CDP          1.356e+00  1.194e-01   11.358   <2e-16 ***
## CO           3.547e-03  5.804e-03    0.611   0.5412
## NOX         -1.851e-02  1.055e-03  -17.550   <2e-16 ***
## year2015     3.514e+00  4.456e-02   78.852   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7339 on 11824 degrees of freedom
## Multiple R-squared:  0.998,  Adjusted R-squared:  0.998
## F-statistic: 5.278e+05 on 11 and 11824 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm1)
```



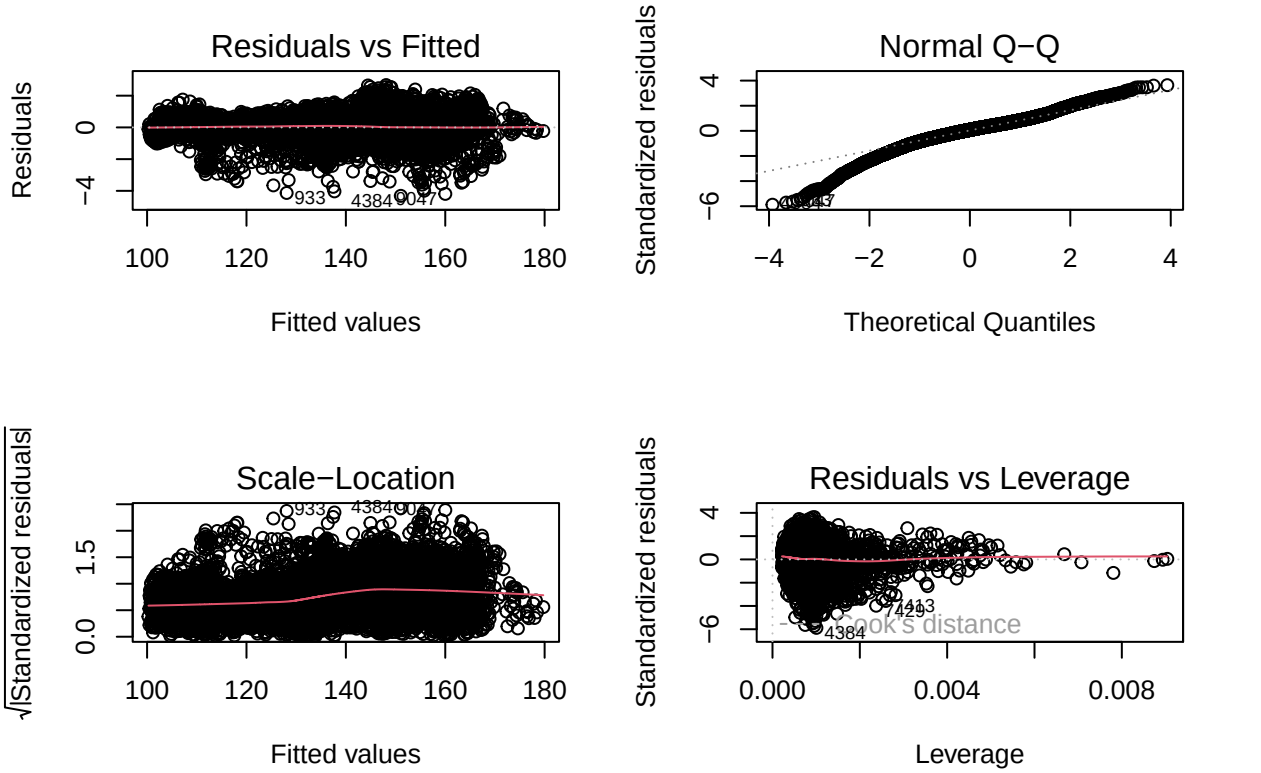




```
lm2 <- lm(TEY~.-CO-AFDP-GTEP, data=train)
summary(lm2)
```

```
##
## Call:
## lm(formula = TEY ~ . - CO - AFDP - GTEP, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3142 -0.3744  0.0594  0.4235  2.6679
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.446e+02  1.669e+00  -86.64   <2e-16 ***
## AT          -3.266e-01  1.606e-03 -203.31   <2e-16 ***
## AP          -6.778e-02  1.239e-03  -54.73   <2e-16 ***
## AH          -1.200e-02  6.599e-04  -18.19   <2e-16 ***
## TIT          6.633e-01  5.656e-03  117.26   <2e-16 ***
## TAT         -6.962e-01  8.322e-03  -83.66   <2e-16 ***
## CDP          1.350e+00  1.193e-01   11.32   <2e-16 ***
## NOX         -1.852e-02  9.178e-04  -20.18   <2e-16 ***
## year2015     3.506e+00  4.140e-02   84.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.734 on 11827 degrees of freedom
## Multiple R-squared:  0.998,  Adjusted R-squared:  0.998
## F-statistic: 7.256e+05 on 8 and 11827 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm2)
```




```
anova(lm1, lm2)
```

```
## Analysis of Variance Table
##
## Model 1: TEY ~ AT + AP + AH + AFDP + GTEP + TIT + TAT + CDP + CO + NOX +
##     year
## Model 2: TEY ~ (AT + AP + AH + AFDP + GTEP + TIT + TAT + CDP + CO + NOX +
##     year) - CO - AFDP - GTEP
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1  11824 6367.8
## 2  11827 6371.1 -3   -3.3387 2.0665 0.1024
```

```
predLR <- predict(lm2, newdata = test)
correlationLR <- cor(predLR, test$TEY)
mseLR <- mean((predLR - test$TEY)^2)
rmseLR <- sqrt(mseLR)
cat("Linear Regression \nCorrelation: ", correlationLR, "\nlm1 Mean Square Error: ", mseLR, "\nlm1 Root
```

```
## Linear Regression
## Correlation:  0.9989558
## lm1 Mean Square Error:  0.5343328
## lm1 Root Mean Square Error:  0.7309807
```

## kNN Regression model

```
library(caret)
```

```
## Loading required package: ggplot2
```

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

```
fit_kNN <- knnreg(cbind(train[,1:7], train[,9:12]), train[,8], k= 3)

predkNN <- predict(fit_kNN, cbind(test[,1:7], test[,9:12]))
correlationkNN <- cor(predkNN, test$TEY)
msekNN <- mean((predkNN - test$TEY)^2)
rmsekNN <- sqrt(msekNN)


cat("kNN Regression \nCorrelation: ", correlationkNN, "\nMean Square Error: ", msekNN, "\nRoot Mean Squa
```

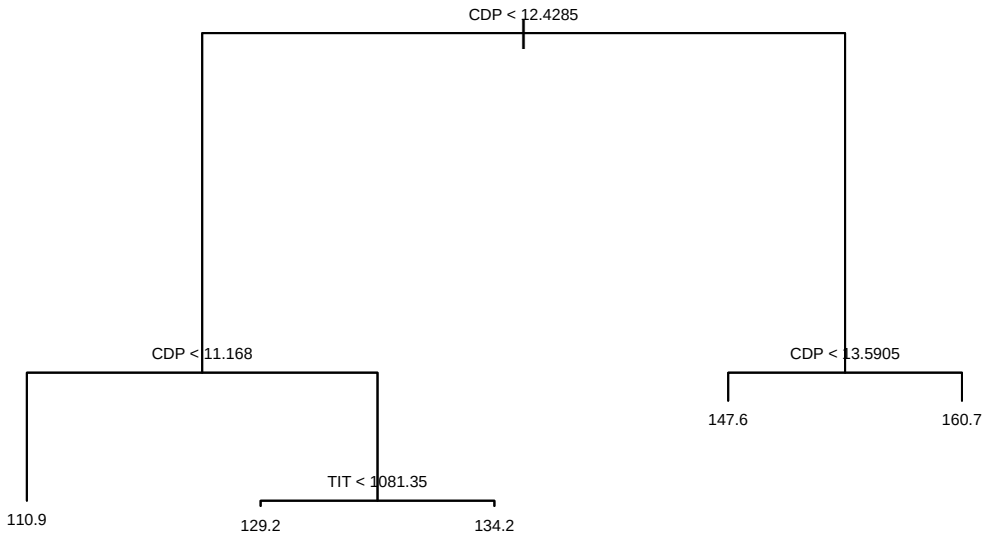
```
## kNN Regression
## Correlation:  0.9971731
## Mean Square Error:  1.446321
## Root Mean Square Error:  1.202631
```

## Decision tree Regression model

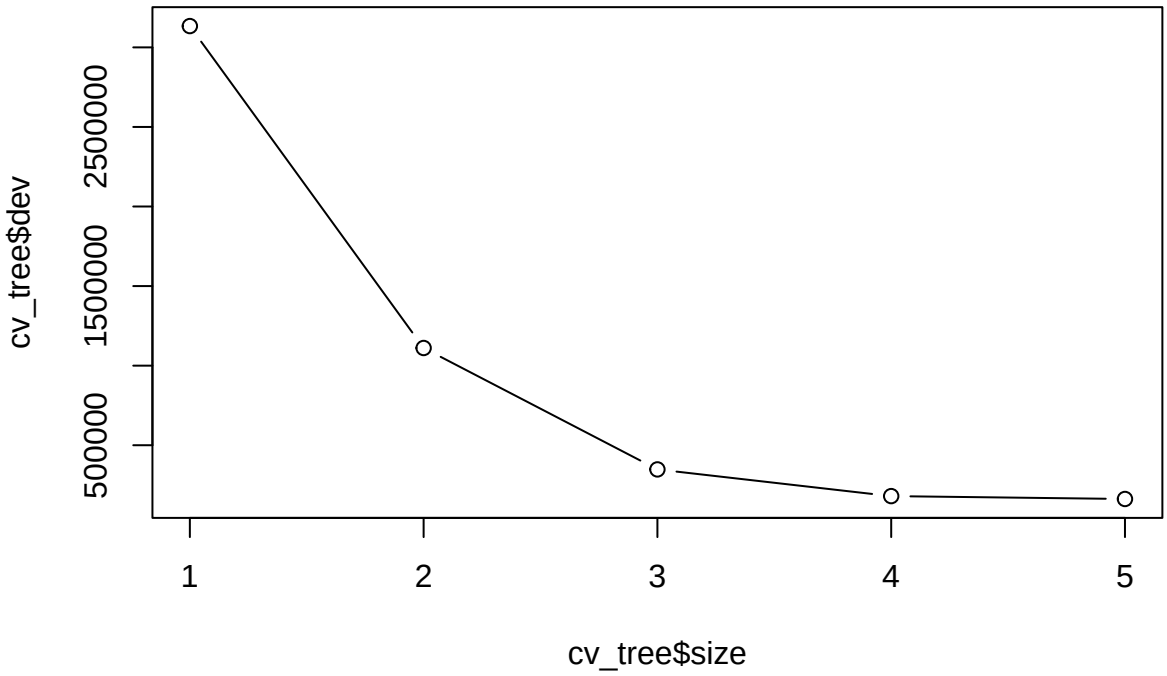
```
library(tree)
tree1 <- tree(TEY~., data=train)
summary(tree1)
```

```
##
## Regression tree:
## tree(formula = TEY ~ ., data = train)
## Variables actually used in tree construction:
## [1] "CDP" "TIT"
## Number of terminal nodes:  5
## Residual mean deviance:  12.43 = 147100 / 11830
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -13.11000  -1.48600  -0.03604   0.00000   1.67400  18.83000
```

```
plot(tree1)
text(tree1, cex=0.5, pretty=0)
```



```
cv_tree <- cv.tree(tree1)
plot(cv_tree$size, cv_tree$dev, type='b')
```




```
predDT <- predict(tree1, newdata=test)
correlationDT <- cor(predDT, test$TEY)
mseDT <- mean((predDT - test$TEY)^2)
rmseDT <- sqrt(mseDT)

cat("\nDecision Tree Regression \nCorrelation: ", correlationDT, "\nMean Square Error: ", mseDT, "\nRoo
```

```
##
## Decision Tree Regression
## Correlation:  0.9764993
## Mean Square Error:  11.91347
## Root Mean Square Error:  3.451589
```

### Analysis of results

Correlation: Of the three models linear regression had the greatest correlation, with the decision tree model having the lowest correlation.

MSE/RMSE: Of the three models linear regression had the least MSE/RMSE, with the decision tree model having the highest.

Still all three of the algorithms have a high correlation and a relatively low MSE meaning they are good models for predicting Turbine Energy Yield. The reason linear regression preformed well on the data set is likely due to the linear relationship between several of the predictors used in the data set and the TEY. kNN preformed well at building a regression model for finding TEY because data points close to eachother in the data set have similar values for TEY. The likely reason for the decision tree model not working as well for the data set is because it overfit the training data leading to a less accurate result when compared to the other two models.