IS624 - Final

James Quacinella

07/12/2015

Contents

Introduction

To quote directly from the data source itself:

> The dataset contains 9568 data points collected from a Combined Cycle Power Plant over 6 years (2006-2011), when the power plant was set to work with full load. Features consist of hourly average ambient variables Temperature (T), Ambient Pressure (AP), Relative Humidity (RH) and Exhaust Vacuum (V) to predict the net hourly electrical energy output (EP) of the plant.
>
> A combined cycle power plant (CCPP) is composed of gas turbines (GT), steam turbines (ST) and heat recovery steam generators. In a CCPP, the electricity is generated by gas and steam turbines, which are combined in one cycle, and is transferred from one turbine to another. While the Vacuum is collected from and has an effect on the Steam Turbine, the other three o the ambient variables effect the GT performance.

In this project, I will try to develop predictive models for this dataset. I will start with linear regression and see what other models can bring.

Data Preperation

The first thing to do was convert the .ods file into a .csv file using LibreOffice. The data.csv file is the result, and has been loaded:

```
# Init
library(e1071)
library(caret)
library(corrplot)
library(ggplot2)
library(GGally)
library(forecast)

# Set random set for predictability
```

```r
set.seed(200)

# Load data
df <- read.csv("CCPP/data.csv")
predictors <-  c("V", "AT", "RH", "AP")
#df.predictors <- df[ , c("V", "AT", "RH", "AP")]
#df.predict <- data.frame(PE=df[ , c("PE")])
```

Lets see if there are any non-complete cases in the data:

```r
sum(complete.cases(df$AT)) == nrow(df) # True
sum(complete.cases(df$V)) == nrow(df)  # True
sum(complete.cases(df$AP)) == nrow(df) # True
sum(complete.cases(df$RH)) == nrow(df) # True
```

It looks like we have a full data set with no missing NA values. Lets see if any of the predictors are significantly skewed:

```r
skewValues <- apply(df[, predictors], 2, skewness)
head(skewValues)
```

```
##          V         AT         RH         AP
##  0.1984588 -0.1363503 -0.4317034  0.2653615
```

From this I would say that there is no significant skewness here, so Box Cox transformations are not needed.

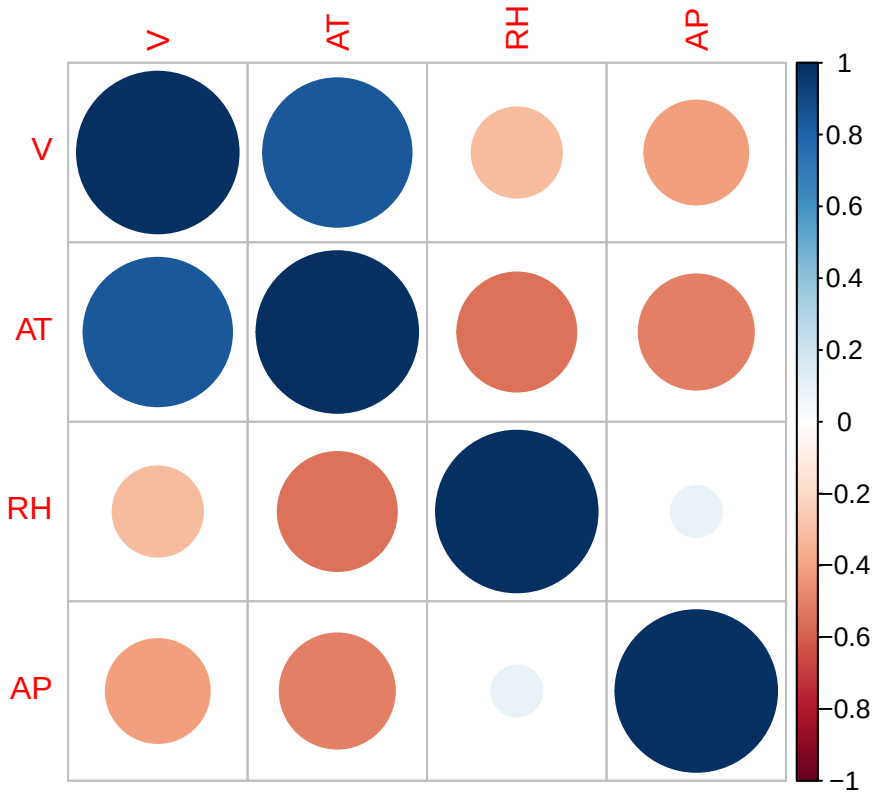
TODO: * should we transform via center and scaling?

Lets remove any near zero variance predictors:

```r
remove <- nearZeroVar(df[ , predictors])   # No columns are near zero variance
```

Are there any correlations between predictors?

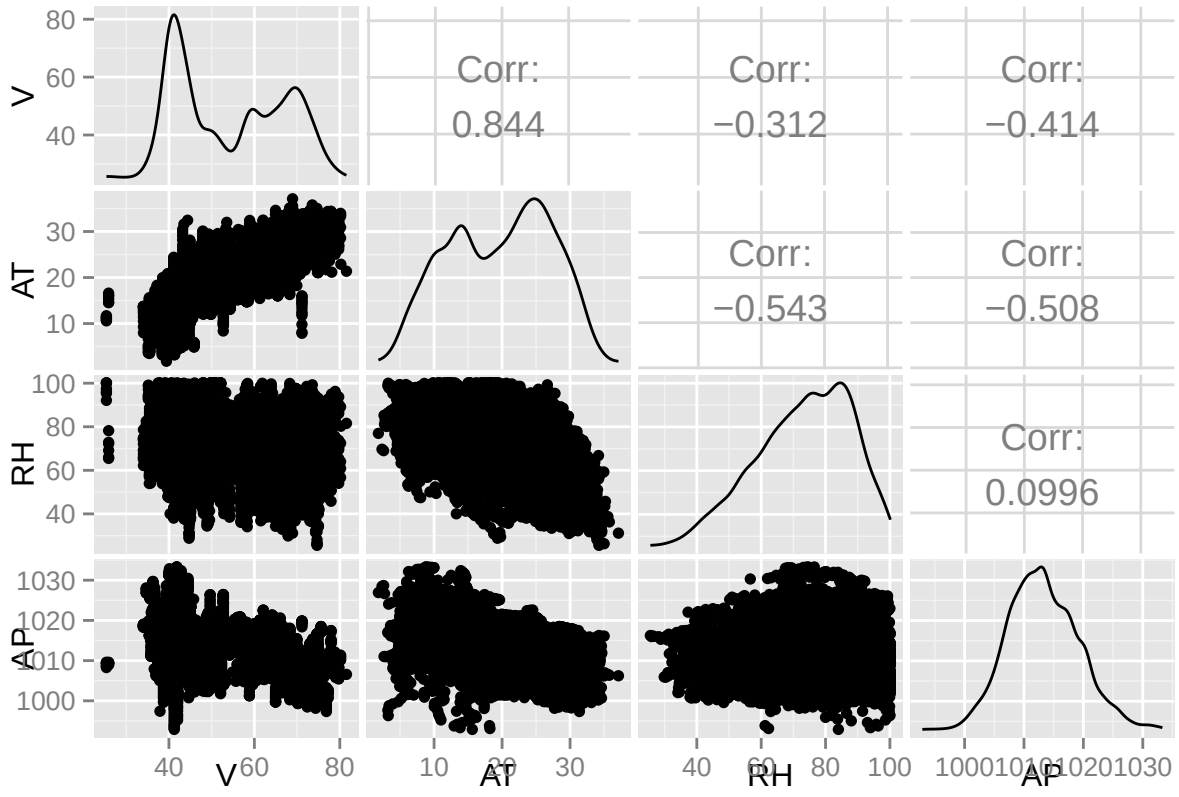
```r
correlations <- cor(df[ , predictors])
corrplot::corrplot(correlations, order = "hclust")
```



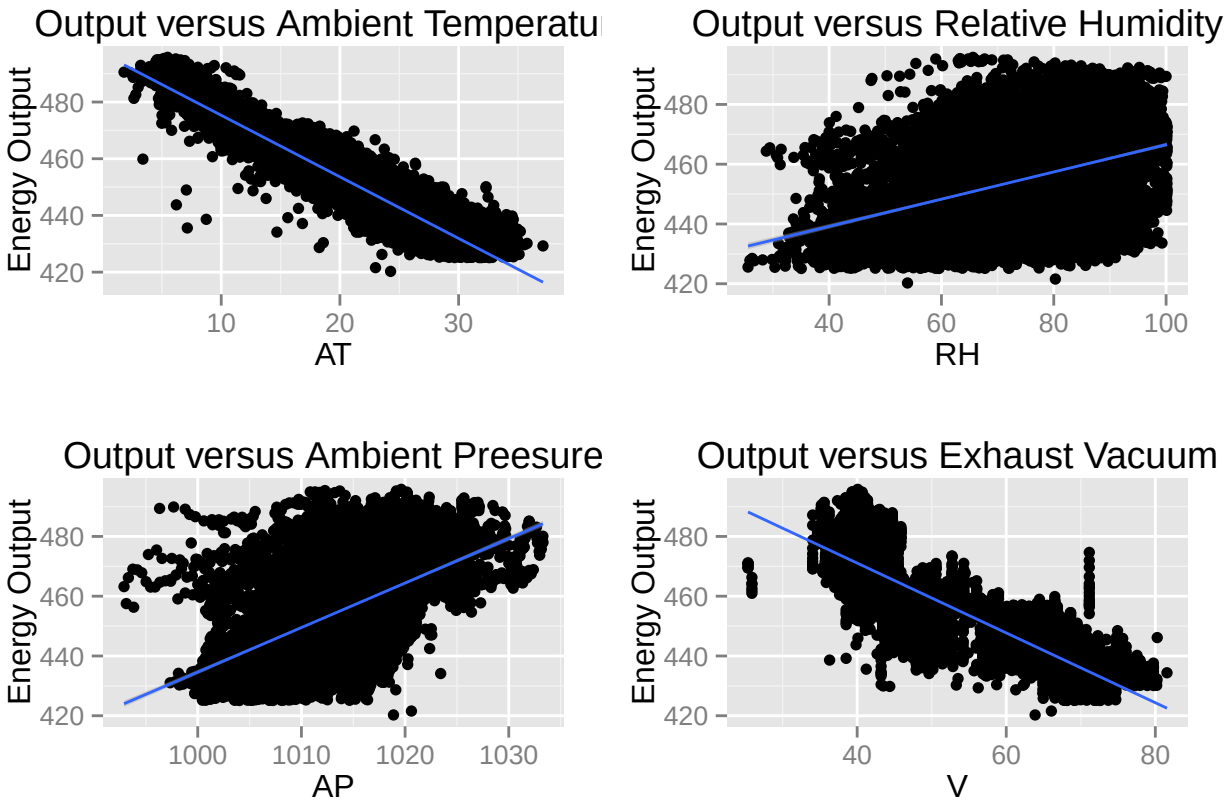
From this we can see some correlations between predictors, but that might be normal: the operating conditions of the plant probably change according to fundamental physical laws and operating contraints, so their changes over time might be correlated.

Lets view that data:

```
ggpairs(df[ , predictors])
```




```
p1 <- ggplot(df, aes(x=AT, y=PE)) + geom_point() + geom_smooth(method='lm') + ggtitle("Output versus Amb
p2 <- ggplot(df, aes(x=AP, y=PE)) + geom_point() + geom_smooth(method='lm') + ggtitle("Output versus Amb
p3 <- ggplot(df, aes(x=RH, y=PE)) + geom_point() + geom_smooth(method='lm') + ggtitle("Output versus Rel
p4 <- ggplot(df, aes(x=V, y=PE)) + geom_point() + geom_smooth(method='lm') + ggtitle("Output versus Exha
multiplot(p1, p2, p3, p4, cols=2)
```



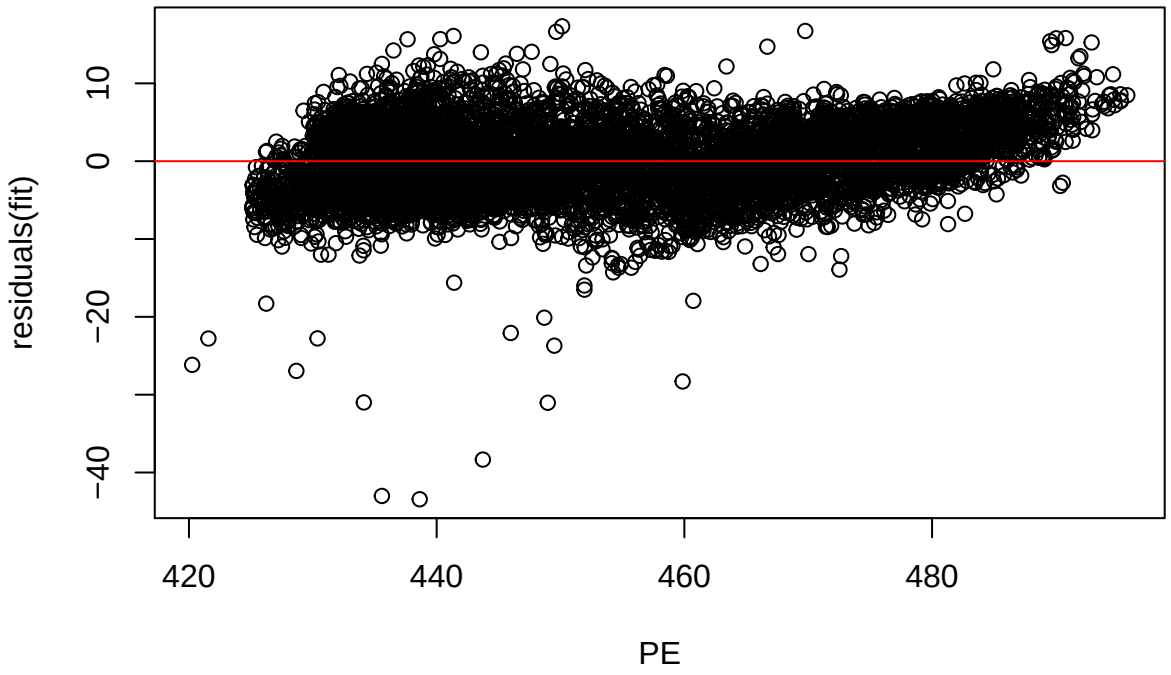
For the next steps, where we build models for this data, lets create a training and test set for performance measures:

```
trainIndex <- createDataPartition(df$PE, p = .8, list=FALSE)
df.training <- df[trainIndex, ]
df.test <- df[-trainIndex, ]
```

Linear Regression Model

```
# Build Lineat Regression Model
fit  <- lm(PE ~ V + AT + RH + AP, data=df.training)
#summary(fit)

# Residuals
plot(residuals(fit) ~ PE, data=df.training, main="Residuals Plot")
abline(0, 0, col='red')
```



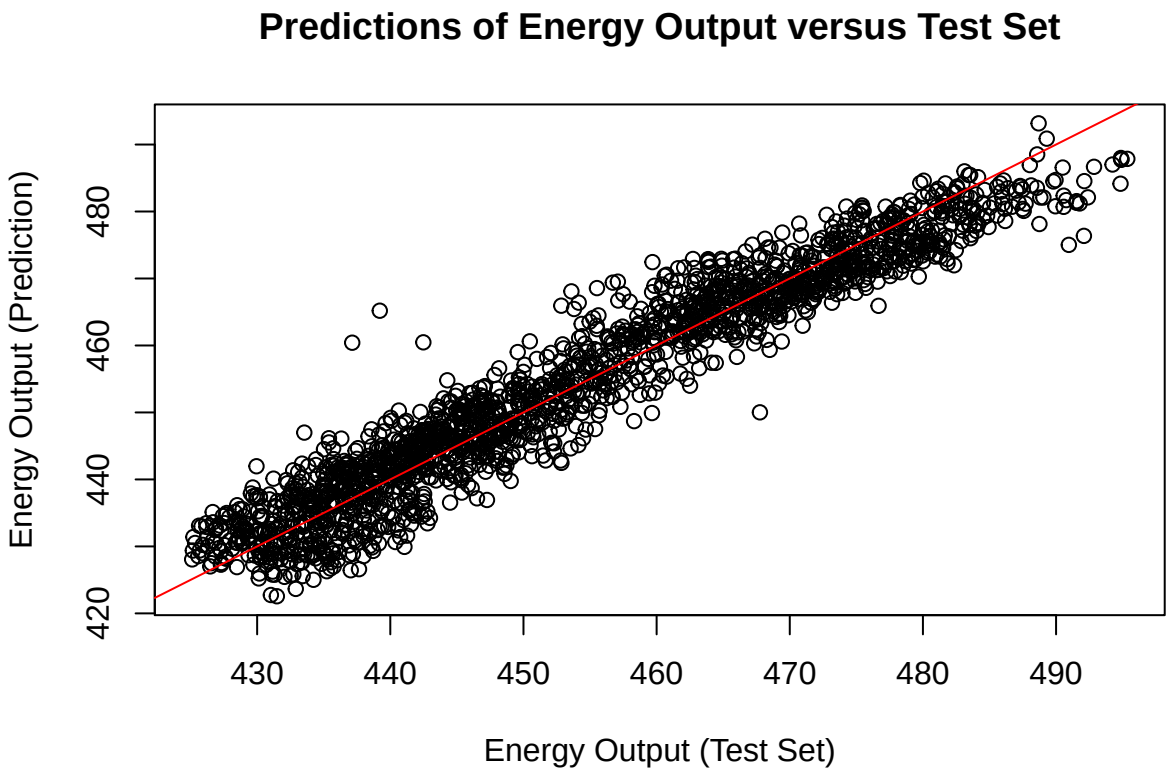

```
# Forecast the data
forecast <- forecast(fit, newdata=df.test)
accuracy(forecast, df.test)
```

```
##                        ME     RMSE      MAE          MPE      MAPE
## Training set  3.152272e-16 4.573928 3.632706 -0.009864752 0.8011648
## Test set     -1.547917e-01 4.490423 3.605224 -0.044617264 0.7952603
##                   MASE
## Training set 0.2445819
## Test set     0.2427316
```

```
postResample(pred = forecast$mean, obs = df.test$PE)
```

```
##      RMSE  Rsquared
## 4.4904226 0.9299941
```

```
# Plot predictions versus reality in test set
plot(df.test$PE, forecast$mean, main="Predictions of Energy Output versus Test Set", ylab="Energy Output
abline(0, 1, col='red')
```



Linear Model with Less Predictors

Lets try taking out one of the highly correlated predictors and see if we can improve our model:

```
# Build Linear Regression Model
fit  <- lm(PE ~ V + RH + AP, data=df.training)
#summary(fit)

# Residuals
#plot(residuals(fit) ~ PE, data=df.training, main="Residuals Plot")
#abline(0, 0, col='red')

# Forecast the data
forecast <- forecast(fit, newdata=df.test)
accuracy(forecast, df.test)
```

```
##                        ME     RMSE      MAE         MPE     MAPE
## Training set -3.444912e-16 7.559133 5.891471 -0.02667179 1.289111
## Test set     -3.741995e-01 7.547288 5.930192 -0.10930903 1.300577
##                    MASE
## Training set 0.3966594
## Test set     0.3992664
```

```
postResample(pred = forecast$mean, obs = df.test$PE)
```

```
##      RMSE  Rsquared
## 7.5472885 0.8023285
```

```
# Plot predictions versus reality in test set
#plot(df.test$PE, forecast$mean, main="Predictions of Energy Output versus Test Set", ylab="Energy Outp
#abline(0, 1, col='red')
```

Taking out the Ambient Temperature predictor, despite having high correlations to the other predictors, does not seem to help here.

More Advanced Models

Results

Citations

Pınar Tüfekci, Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods, International Journal of Electrical Power & Energy Systems, Volume 60, September 2014, Pages 126-140, ISSN 0142-0615, http://dx.doi.org/10.1016/j.ijepes.2014.02.027. (http://www.sciencedirect.com/science/article/pii/S0142061514000908)

Heysem Kaya, Pınar Tüfekci , Sadık Fikret Gürgen: Local and Global Learning Methods for Predicting Power of a Combined Gas & Steam Turbine, Proceedings of the International Conference on Emerging Trends in Computer and Electronics Engineering ICETCEE 2012, pp. 13-18 (Mar. 2012, Dubai)