Stat 508 - Final Project

Adam Behrendorff
Nicholas Napier
Saqib Ali

Contents

```
suppressMessages(library(lubridate))
suppressMessages(library(caret))
suppressMessages(library(corrplot))
suppressMessages(library(sugrrants))
suppressMessages(library(dplyr))
suppressMessages(library(MASS))
suppressMessages(library(e1071))
```

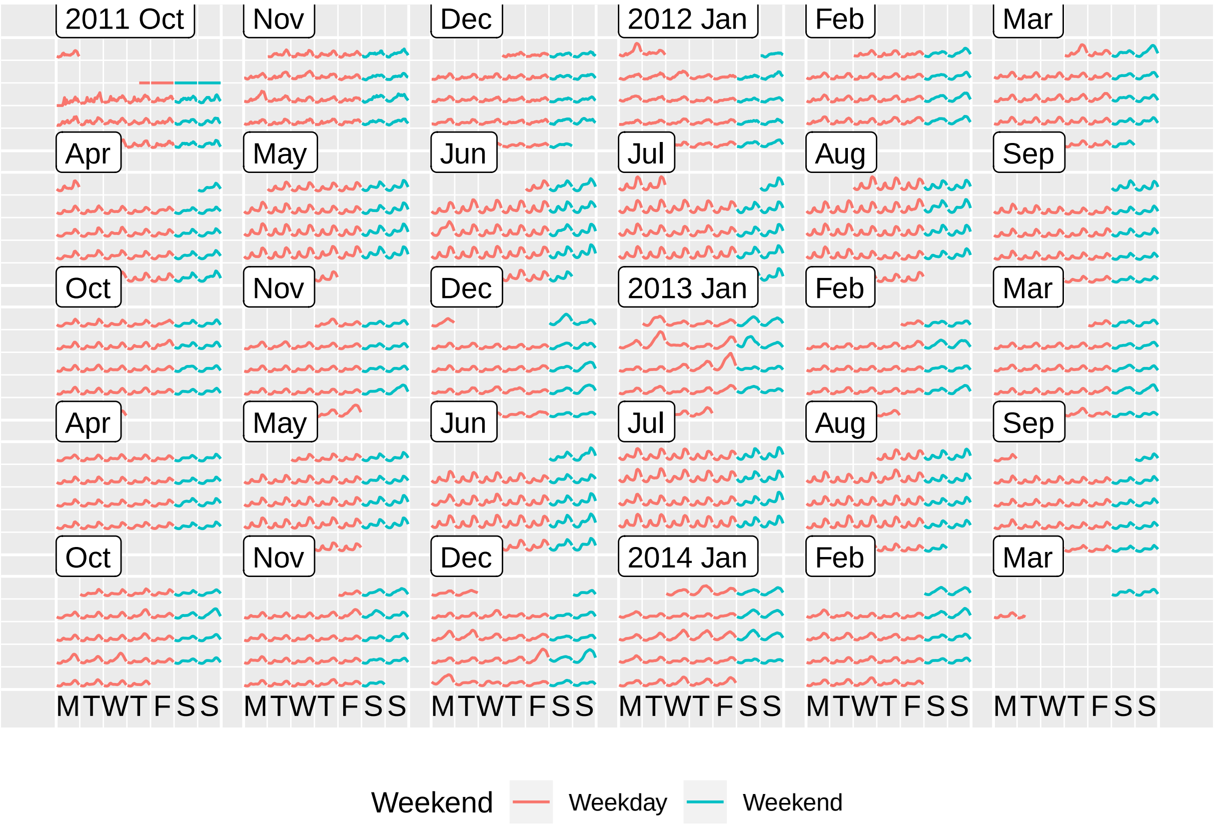
Introduction

Data

```
EnergyDataAggregate <- read.csv("EnergyDataAggregate.csv",stringsAsFactors = FALSE)
EnergyDataAggregate$DATE <- as.Date(EnergyDataAggregate$DATE,format="%m/%d/%Y")
EnergyDataAggregate$Weekend <- if_else(EnergyDataAggregate$DAYNAME %in% c("Saturday", "Sunday"), "Weeken
```

Time-series Trend

```
p <- EnergyDataAggregate %>%
  frame_calendar(x = HOUR, y = GENERAL_SUPPLY_KWH, date = DATE) %>%
  ggplot(aes(x = .HOUR, y = .GENERAL_SUPPLY_KWH, group = DATE, colour = Weekend)) +
  geom_line() +
  theme(legend.position = "bottom")
prettify(p)
```



Analysis

Binary Response Variable

Since we working on predicting a High Consumption vs. Low Consumption, we will make a HighEvergyUse binary variable based on the mean GENERAL_SUPPLY_KWH for the hour in the day

```
EnergyDataAggregate$HighEnergyUse <- ifelse(EnergyDataAggregate$GENERAL_SUPPLY_KWH>mean(EnergyDataAggre
EnergyDataAggregate$MONTH <- as.factor(EnergyDataAggregate$MONTH)
EnergyDataAggregate$DAY <- as.factor(EnergyDataAggregate$DAY)
```

Spliting the Dataset

We will split the dataset into training and testing splits. We will use all the data from before 2013 to build a model to predict Evergy Consumption for years 2013 and higher.

```
set.seed(1)

#training and test set
energyData.full <- EnergyDataAggregate
energyData.train=EnergyDataAggregate[EnergyDataAggregate$YEAR<2013,]
energyData.test=EnergyDataAggregate[EnergyDataAggregate$YEAR>2013,]
```

Logistic Regression

Let's build a Logistic Model using the training Data

```
logit.fit <- glm(HighEnergyUse~Weekend+DAY+MONTH+DAYNAME+HOUR, data=energyData.train, family=binomial)
```

```
glm.probs=predict(logit.fit,energyData.test,type="response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
## == : prediction from a rank-deficient fit may be misleading
```

```
glm.pred <- ifelse(glm.probs<0.5, 0, 1)

pred.mean <- mean(glm.pred==energyData.test$HighEnergyUse)
pred.mean
```

```
## [1] 0.8009352
```

We observe that with Logistic Regression, we get and accurary rate of 0.8009352

LDA

```
lda.fit=lda(HighEnergyUse~Weekend+DAY+MONTH+DAYNAME+HOUR, data=energyData.train)
```

```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

```
lda.pred=predict(lda.fit, energyData.test)
pred.mean <- mean(lda.pred$class==energyData.test$HighEnergyUse)
```

We observe that with LDA, we get and accurary rate of 0.8056112

QDA

With Day of the Month

```
qda.fit=qda(HighEnergyUse~DAY+MONTH+DAYNAME+HOUR, data=energyData.train)
```

```
qda.pred=predict(qda.fit, energyData.test)
pred.mean <- mean(qda.pred$class==energyData.test$HighEnergyUse)
```

We observe that with QDA, we get and accurary rate of 0.6726787

Without Day of the Month

```
qda.fit=qda(HighEnergyUse~DAY+MONTH+DAYNAME+HOUR, data=energyData.train)
```

```
qda.pred=predict(qda.fit, energyData.test)
pred.mean <- mean(qda.pred$class==energyData.test$HighEnergyUse)
```

We observe that with QDA, we get and accurary rate of 0.6726787

Support Vector Classifer

```
#tune.out=tune(svm, HighEnergyUse~DAY+MONTH+DAYNAME+HOUR,data=energyData.full,kernel="linear",ranges=li
```